DTU Library

# Navigating Distributed Services

**Beute, Berco**

*Publication date:*
2002

*Document Version*
Publisher's PDF, also known as Version of record

CTI Ph.D. series, no. 3

# Navigating Distributed Services

**Doctoral thesis**
**Berco Beute**
Center for Tele-Information
Technical University of Denmark
Supervisor: Associate Professor Michael Rose
Funding: CIT
Project affiliation: Multimedia in the Home (125)
Kgs. Lyngby
September 2002

Published by Center for Tele-Information, Technical University of Denmark, Kongens Lyngby, 2003.



Printed by Schultz DocuCenter, Denmark

ISBN 87-90288-15-7
ISSN 1601-720X

“The lurking suspicion that something could be simplified is the world’s richest source of rewarding challenges.”

Edsger Wiebe Dijkstra (1930-2002)

# Acknowledgements

First and foremost I would like to thank my supervisor Michael Rose for valuable insights, creative solutions, and continuing support throughout the process of writing the thesis.

Without the many discussions with Jakob Eg Larsen my thesis would have been a lot less interesting. His thorough understanding of the topic, his visionary approach, and his amiable cooperation made the whole process a lot more pleasant.

Mark Paul Saunders deserves my gratitude for excellent proofreading, useful editing and humorous feedback.

Regarding my stay at ICT@NN in The Netherlands I would like to thank Kees Nieuwenhuis for his help and inspiring feedback.

Thanks to everybody working at the Center for Tele-Information for the pleasant time you gave me in Denmark. Special thanks for this go out to Henning Olesen and Kristian Kristensen.

Finally, I am most grateful to the one to whom I not only dedicate this thesis, but my life: D'or.

Berco Beute

Groningen, The Netherlands

2002

# Abstract

This thesis explores the impact of three current trends which, when taken together, are fundamentally changing the way in which the task of navigating virtual environments is accomplished. The first concerns the changeover from a situation in which all data and functionality reside locally to the user, to a situation where they are distributed across the Internet. The second trend is the shift from a virtual environment that solely consists of distributed documents to a virtual environment that consists of both distributed documents and distributed services. The third and final trend is the increasing diversity of devices used to access information on the Internet.

The focal point of the thesis is an initial exploration of the effects of the trends on users as they navigate the virtual environment of distributed documents and services. To begin the thesis uses scenarios as a heuristic device to identify and analyse the main effects of the trends. This is followed by an exploration of theory of navigation Information Spaces, which is in turn followed by an overview of theories, and the state of the art in navigating distributed services. These explorations of both theory and practice resulted in a large number of topics for further investigation.

The thesis focuses upon three sub-topics. The first deals with the general differences between navigating distributed documents and navigating distributed services. The second deals with the applicability of a geographical metaphor for collections of distributed services. The third and final sub-topic tries to answer the question of the different metadata requirements distributed services have when compared to distributed documents.

A study is devised to test the validity of three hypotheses, but also to provide specific details about the differences in how users search for documents vs. services, and to give a detailed overview of the required metadata for services. The study includes the building of prototypes that are evaluated by experts and tested by users.

Together, the scenario design, the literature review, the building of the prototype, and expert and user evaluations test the validity of the hypotheses, the results of which can be used to improve the user experience of navigation distributed services. Furthermore the results include both a fully functional platform for browsing services and a large number of services. Both can be used for further studies.

# Abstract – Danish

Denne afhandling undersøger betydningen af tre tendenser, der tilsammen medfører fundamentale ændringer i, hvordan virtuelle miljøer styres (navigeres). Den første tendens vedrører overgangen fra en situtation, hvor al data og funktionalitet befinder sig lokalt til en situation, hvor de er distribueret over Internettet. Den anden tendens er overgangen fra et virtuelt miljø, som udelukkende udgøres af virtuelle dokumenter, til et virtuelt miljø, der består både af distribuerede dokumenter og distribuerede tjenester. Den trejde, og sidste tendens, er den stigende mangfoldighed af udstyr, som benyttes for at få adgang til information på Internettet.

Afhandlingen benytter scenarier som et heuristisk redskab til at identificere og analysere hovedeffekterne af tendenserne. Fokus i denne afhandling er, som nævnt, hvordan ovennævnte tendenser påvirker brugere, idet de navigerer i det virtuelle miljø af distribuerede dokumenter og tjenester. Derefter undersøges teorien for navigation i Information Spaces. Efterfølgende gives et overblik over teorier og state of the art i navigation i distribuerede tjenester. Disse undersøgelser af både teori og praksis resulterede i et større antal områder for videre forskning.

Der fokuseres på tre under-emner. Det første vedrører de generelle forskelle mellem at navigere distribuerede dokumenter og at navigere distribuerede services. Det andet vedrører anvendeligheden af en geografisk metafor for samlinger af distribuerede tjenester. Tredje, og sidste under-emne, søger at besvare spørgsmålet om de forskellige metadata krav til distribuerede tjenester til sammenligning med kravene for distribuerede dokumenter.

En undersøgelse til besvarelse af spørgsmålene forbundet med denne forskning er udført. Undersøgelsen indbefatter opbygningen af en prototype, som er evalueret af eksperter og afprøvet af brugere.

Scenarie-designet, litteraturgennemgangen, opbygningen af en prototype, ekspert- og bruger-evalueringerne udgør tilsammen kvalitative svar på forskningsspørgsmålene. Svarene kan anvendes til at forbedre brugerens erfaring i navigation i distribuerede services. Yderligere omfatter resultaterne både en funktionel platform til browsing ydelser og et stort antal tjenester. Begge dele kan udforskes yderligere.

# 1. Introduction

*"The times they are a' changing", Bob Dylan sang back in the nineteen sixties. He was just as right about the cultural situation of his time as he would have been about the situation of using computers these days. After exactly two decades of personal computers the industry stands at the dawn of what might turn out to be the biggest revolution it has seen until now. Three trends slowly change the way in which everyday computer users will search for, find, and consume, everything and more that computers offer today. The first trend is going from a local to a distributed setting, the second trend is going from documents to services and the third trend is using multiple devices instead of one personal computer for accessing the services and documents. With new possibilities come new questions, some of which will be answered in this thesis. For this the trends need to be explained in more detail and an overview of the issues they raise should be given, which will be done in this chapter. Subsequently this chapter delimits the problem domain followed by a brief overview of the relevant research areas and topics. This will lead to the statement of the hypotheses and their validity. The chapter concludes with an outline of this thesis.*

Perhaps one of the biggest frustrations of the last two decades has been the personal computer. Most readers can probably come up with excellent examples of how unpredictable such a machine can be. It's no coincidence that Donald Norman - one of most renowned usability experts, closed the millennium with a book that has '*Why Personal Computers Are So Complex*' as its title[1]. After twenty years the personal computer has turned into a seemingly magical machine that can do almost anything. But this comes with a price; it suffers from what sometimes is being referred to as the 'Swiss army-knife effect'. By trying to do so many things, from clipping nails to sawing wood, a Swiss army knife fails to be good at any of them. In particular what the knife initially was meant to do well, namely cutting. Personal computers started off as machines that let you type letters but soon programmers invented many more things it could do. The personal computer's functionality has increased up to the point where it can now be used to call your parents, write a letter, watch television, create paintings, make music, browse an unlimited number of documents on a network and turn the light on. This increase in functionality comes with a price - increased complexity (Wack, 1985;Gelernter, 1998;Segal, 1995;Cooper, 1995;Mohnkern and Carnegie Mellon University.School of Design., 1997;Sutcliffe, 1995;Mayhew, 1992;Laurel, 1995). To make the personal computer support these diverse tasks a user interface had to be created for each task that fitted the characteristics of a computer and that supported the task. This required many decisions where usability had to compete with aspects such as screen-size and memory usage, and often lost. Not only

---

[1] Full title: "The invisible computer: Why good products can fail, the personal computer is so complex and information appliances are the solution" (Norman, 1998).

were the individual applications harder to use than the original tools[2], the entire set of applications required more and more time and attention to administer.

The first personal computers came with an operating system and a few applications, but pretty soon the market was flooded with applications users could install themselves. The challenge of getting these applications to work smoothly side by side, (and sometimes together,) on the one machine turned users into system administrators for their own personal computers. This job became increasingly complex. Paradoxically most users didn't use their personal computer for all these things, they mainly used it to do a limited set of tasks, such as word processing and sending e-mail (Norman, 1998). Thus, at the close of the millennium, the awkward situation is that a large part of western society has a machine in their homes that can *do* a many things but is bought for one or two specific tasks for which it isn't used - because that is too complex. It is too complex because it can do so many things. It seems that this vicious circle of ever increasing complexity cannot be broken.

Fortunately there is light at the end of the tunnel. At the time of writing there are three trends (mentioned earlier) that seem to make accessing and using computer functionality yet more difficult, but they could very well turn out to be opportunities to remove technological hurdles allowing users to once more focus upon the task at hand. At present each trend looks rather obvious in and of itself, it is in combination that they may hold an answer to a more user-friendly future. In the next section each trend will be discussed more thoroughly. Thereafter a small scenario will be worked out and analysed to introduce the research field.

Before continuing it should be stressed that the scope of this research is the near future (next five to ten years). Beyond that range detailed predictions about information technology are impossible. Numerous examples of false predictions can be found[3], for example the chairman of IBM (Thomas Watson) in 1943 saying, "I think there is a world market for maybe five computers".

## 1.1. A tale of three changes

### 1.1.1. Change 1: From a local to a distributed setting

The first change was initiated by the coming of the Internet and its most popular application - the World Wide Web (Web hereafter). The Web was invented to allow scientists to author documents on each others computers (Berners-Lee, 1989;Berners-Lee, 1989). The reason for this was preventing duplication of the same document on computers around the world. Working on one document with different persons was

---

[2] Compare for instance the use of a computer keyboard with an electric piano that has MIDI capabilities for making computer music.

[3] Examples can be found here: http://rinkworks.com/said/predictions.shtml

quite difficult and prone to errors with various versions of the document on different computers. The story of the Web from there on is well-known[4]. With the penetration of the Web into the fabric of everyday life users got accustomed to storing their documents somewhere on the Internet, rather than locally on their personal computer. Not just the obvious content such as things they wrote themselves but also books, folders, yellow pages, dictionaries etcetera were stored on the Internet.

A side effect of this distribution trend was that it became more difficult for users to find their applications and data (Theng, Thimbleby, and Jones, 1996;Horton, 1994;Thimbleby, Jones, and Theng, 2002). When both documents and applications were local it was rather easy[5] to find something due to the limited number of applications and documents that could be stored on a standalone personal computer. When both the documents were distributed across the Internet it became a lot more difficult to find the right document. It seemed as though all documents in the world were thrown onto one large heap, making the task of finding a document a lot more difficult.

### 1.1.2. Change 2: From documents to services

Through the last decade we saw a slow rise in functionality being offered *via* the Internet (Norman, 1998). People started using the Internet to play games, watch online streaming videos, calculate their expenses, program their VCR, or make phone calls while looking at a live picture of the other party. Soon they will be able to connect to their oven to see if the cake is ready. Slowly but surely the Internet became a medium that offered more than just document sharing. It has become a medium that delivers functionality as well (McIlraith and others, 2001, 46-53). It was still very limited what could be delivered though, the physical network as well as the protocols and applications on top of the Internet were not ready for delivering all kinds of functionality (Segal, 1995). It wasn't until the introduction of broadband Internet connections in the late nineteen nineties that a somewhat more sophisticated functionality could be consumed through the Internet. The first pieces of functionality required little bandwidth[6], things like Java applets[7] for calculating an exchange rate only required a small amount of data to be transported across the Internet. The Internet, having grown in capacity with the increasing popularity of the Web, was well suited for that task. But users got used to the fact that more could be done with the Internet and started to demand more functionality to be accessible *via* the Internet. The industry reacted by providing users with more functionality, and soon users were able

---

[4] If not, read Tim Berners-Lee's own excellent account of it (Berners-Lee and Fischetti, 1999).

[5] Pun intended.

[6] The amount of data a network is able to transport within a certain time.

[7] Java is a programming language and a Java applet is a program that is downloaded from a server and run inside a browser.

to chat with others, watch videos and listen to music through the Web browser. One could argue, after the dotcom meltdown, that the industry's reaction was a little overheated, but that's a topic outside the scope of this thesis. At least the hype temporarily secured an almost bottomless supply of investments. During the hype everything below[8] the Internet - the physical network, as well as everything above it - the protocols and applications, were upgraded to be capable of delivering what the consumer apparently craved. Soon the user was able to read e-mail, or watch small videos on demand *via* the Internet. Functionality that was hitherto only available as installed applications on standalone personal computers now was being offered *via* the Internet as well.

It's hard to describe the 'units of functionality' available *via* the Internet in terms of 'applications' because that term is closely related to software running on desktop computers. Fortunately there is a term that better fits the idea of distributed units of functionality; service. The dictionary definition of a service is "work done for others as an occupation or a business" (American Heritage, 1996). In light of this research we are only interested in digital services that come across the Internet, so the working definition of a service as used in this thesis will be[9]:

> ***Service****:*
> *The offering to solve a task made available through a digital network.*

Even within the limited domain of digital networks it is a broad term. It can be used for services as small as turning on a light to as big as a large enterprise IT systems. Furthermore the definition doesn't specify who provides the service, it doesn't matter whether it is a super computer or a mathematical genius offering a calculation service, as long as both are accessible through a digital network. The reason for using the term 'digital network' instead of 'Internet' is that the latter is defined as a network of networks (American Heritage, 1996), which hides the differences between the individual networks. The term 'digital network' includes all types of networks that transport digitised data, whether the network is wireless, fixed, slow or fast[10].

The advantage of describing entities on the network in terms of services is that it abstracts away the difference between whatever it is that provides that service, be it a piece of software, a device, a server or something yet to be invented. From a user

---

[8] Below/above in terms of the OSI model of network communication. See the appendix for an overview.

[9] Every use of the term 'service' throughout the thesis from here onwards refers to this definition unless specified otherwise.

[10] A few examples of networks: GSM, BlueTooth, Wireless LAN, UMTS, Ethernet, infrared connections, GPRS etcetera.

perspective the service is in the end what she is after, and the user perspective is what we're trying to keep to during this research. During a recent conference Jim Waldo gave a very clearly explanation of the service perspective (Waldo, 2001):

> *"Services are an idea that says what you go looking for is not a machine or a particular thing, but something that provides a service for you. Essentially, it is a way of doing late binding -- you don't go out looking for the particular thing that you know does this for you. [Instead] you go looking for whatever it is that you want done."*

It makes more sense to say that a television that is connected to the Internet provides the service of displaying a video instead of saying that it offers an application. The latter might confuse a reader familiar to installing applications on a desktop computer into thinking that she can download an application from the television and install it.

### 1.1.3. Change 3: From one personal computer to multiple devices

The third and last important change is that the number of ways of accessing the collection of information (and services) has grown dramatically (Eisner Gillett and others, 2000;Ishii and Ullmer, 1997;Norman, 1998;Ohta and Tamura, 1999;Rischpater, 2001;Weiser, 1991). Besides a desktop computer users can now use a mobile phone, a personal digital assistant (PDA) or just a microphone with voice recognition software to access online content. The desktop computer is loosing its hegemony for accessing online information. At this moment the company with the largest stake in desktop computers[11] is struggling to get a hold on the device market. They realize that the desktop computer will only be a *part* of the future so to maintain their market share they are making their products available on devices as well. At the same time iMode[12] is being introduced in Europe, the market is flooded with phone/PDA combinations and an increasing number of device manufacturers is showing products that include a network connection. Together these examples indicate that there is an increase in the number of ways information and services can be accessed.

A consequence of the change is that the *same* information or service can be accessed from a wide range of devices. It is no longer the case that a Web page can only be viewed using a desktop computer running a Web browser or that the television can only be controlled with standard remote control. The user might demand to have access to her documents and services from many different devices (Norman, 1998).

[11] Microsoft.

[12] A proprietary architecture for accessing online information and services using a special mobile phone.

Jakob Nielsen (Nielsen, 2000a) noted the same, saying "the new coordinating layer[13] will manage users' access to information objects and functionality objects across multiple devices".

This will in turn effect *what* is being accessed. The initial failure of WAP[14] phones in Europe (Nielsen, 2000b) is a good example in this respect. The first generation WAP phones had very small screens that could display at most a few lines of text. The content that the providers wanted to show however was regular Web pages, but that didn't fit resulting in usability wise inferior products (Ramsay and Nielsen, 2000). In an attempt to solve the problem the phone manufacturers tried to convince the content providers to create special pages for WAP phones, meaning that the content had to adapt to the device. Some attempts were made but it was too little and too late. It wouldn't be until WAP phones with larger displays became available that the protocol would finally be somewhat useful for some users.

The shift from standalone desktop computers to networked information devices for accessing information and services seem to be part of a larger trend where the *environment* becomes the user interface. Such interfaces are sometimes called 'environmental interfaces' (Burkey, 2000;Kindberg, Barton, and Jeff Morgan, 2000) or 'augmented reality' (which I will use here) (Azuma, 1995, 355-385;Ishii and Ullmer, 1997), which is part of the research field called 'ubiquitous and pervasive computing'. Where most of the research in information navigation and visualisation at the beginning of the nineteen nineties focused on humans entering a virtual world (see for instance (Card, Mackinlay, and Shneiderman, 1999)), ubiquitous/pervasive computing took the augmentation of the user's physical surrounding with computing devices as the main focus point. At this moment both trends seem viable, a virtual environment that people can navigate as well as a reality augmented with a digital component. It depends on the task at hand as to which solution better fits the user's needs, but the shift from desktop computers to devices to access information and services can be seen as part of the trend towards an augmented reality.

A side effect of the increase in networked information devices is that those devices can, besides being used to *access* information and services, also *provide* information and services. For instance a desktop computer used for navigating the Web might also offer the service of displaying videos to the network.

---

[13] The layer predicted by hypertext theory to emerge as the new nexus of the user experience (Nielsen, 2000a).

[14] Wireless Access Protocol. A standard for providing information appliances with secure access to text-based Web pages.

## 1.2. General issues

The combination of the three changes raises many issues. A small scenario from a possible future might clarify the consequences of the changes. More detailed scenarios and analyses will be the subject of subsequent chapters.

> This morning Richard worked on a document using his laptop, but while driving to office he realized he omitted something. Not being able to use a screen or keyboard he used the speech interface of the car's computer system to reopen his document and launch his favourite text editor in speech mode. He needed to include some status information from one of the machines on the factory floor. He navigated to the machines service and requested the status. Using speech recognition he included that in the document. Finally he opened the printer service from his office to print the document. When he arrived at the office the document was waiting in

The first set of questions this scenario raises is about navigation. How can Richard find his documents as well as the machine/print services from the home as well as from his car? What does that mean for the tool he uses to navigate? If he uses some kind of browser, what would it look like and how should it work? What does this mean for the entities being navigated? How do they advertise their qualities so that either the user, or another service, can make a better decision? And what about navigation behaviour? With services, does it make sense to use different metaphors from those used for navigating computers? Services might be used in different situations than information, would that change navigational requirements?

The differences between navigating information and navigating services is also noted by Web usability expert Jakob Nielsen (Nielsen, 2000a). In one of his newsletter he mentions, "hypertext theory has predicted the emergence of a navigation layer that would be the nexus of the user experience". He goes on to admit he always assumed that new nexus would be the convergence of the Web browser with the operating system, but he now realizes that the browser is not fit for offering 'information objects' since "application functionality requires more UI than document browsing" (Nielsen, 2000a).

Another topic is security. In the scenario Richard opened a document and some services from different places, but how does he authenticate himself? And how is he authorized to work on that document? With services security becomes more important because there is possibly more at stake, for instance when controlling a heart monitor from a remote location.

Personalization will be affected by the changed situation. Services have different personalization requirements and capabilities than documents, raising new questions: where are the preferences[15] to be stored? How much personal information does Richard allow the services to read? And how does he specify which information can be read by a service and which not? How can users be sure their privacy is protected?

There are technical issues to be addressed as well. One is how a service can present itself across a wide range of devices. In the scenario Richard used a text editor from a laptop at home as well as from the onboard computer in his car. He would like that to be the same service since he is used it, but how can that service be available both on a laptop and *via* a speech interface? There are many possible solutions for that, but the question is what the optimal solution will be.

Another, somewhat different, technical problem underlying the scenario is maintaining this ever-growing collection of interconnected pieces of hardware and software. While this ought to be a job for system administrators it will affect end users as well since users have been forced to act as administrators for their personal computers over the past years. A future such as that depicted in the scenario can only become reality when some of the maintenance burden is lifted from the shoulders of the users.

Yet another technical question is how services and documents will be able to cooperate with as little as possible intervention from the user. Richard accessed different services and used one of them for printing his document. How can such a spontaneous cooperation be achieved?

Finally questions are raised about whether network limitations are being taken into account. Since every service and document Richard uses comes across the network it is important that a possible failure of the network doesn't cause too much damage[16]. Besides that limited bandwidth and latency, just to name two, will affect how Richard experiences the use of services across a network.

## 1.3. Problem domain

The ultimate motivation behind this research is to make it as easy as possible for users to find and use distributed services. The problem is that this goal can be approached from many different angles. One approach would be to focus on the users and their interaction with the system, which is the area of Human-Computer Interaction. Another approach would be trying to build a system that enables the user to accomplish her task, which is the area of Distributed Computing and Ubiquitous Computing. Yet another approach would be to focus on the content that is being

---

[15] Options in a service that can be changed by the user.

[16] This is also called 'graceful degradation', meaning that a system keeps working (although with reduced functionality) after some of its components failed.

navigated, which is the area of Knowledge Technologies. These areas are not completely separated; they have overlapping areas and many interdependencies. This research is based at the convergence of those four broad research areas.

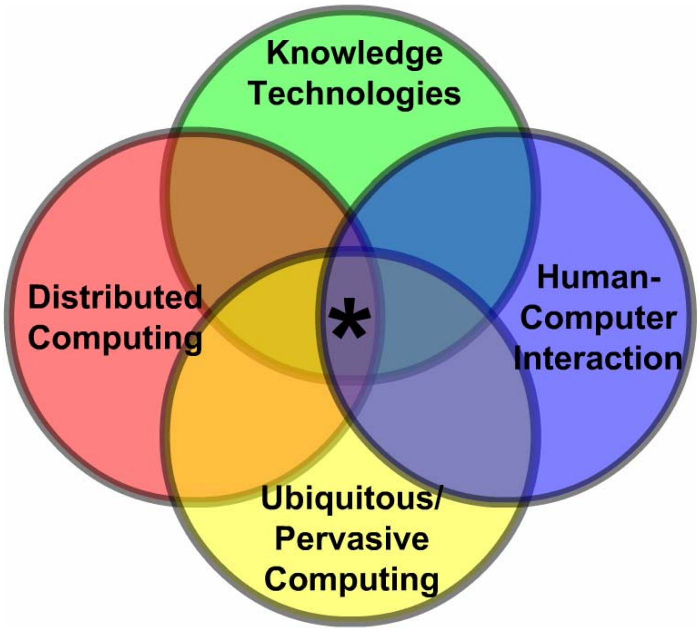



**Figure 1-1: The relevant research areas (* is where this research is located).**

## Human-computer interaction

One way to simplify finding and using services is to look at the user and investigate how she interacts with a system that enables her to do that. A clear understanding of this interaction will help in designing user-friendly computer systems that assist the user in her activities. The following illustration from the SIGCHI[17] committee gives a good overview of all aspects of use and context that are part of the research area of human-computer interaction.

---

[17] Special Interest Group for Computer-Human Interaction (Association for Computing Machinery, ACM)

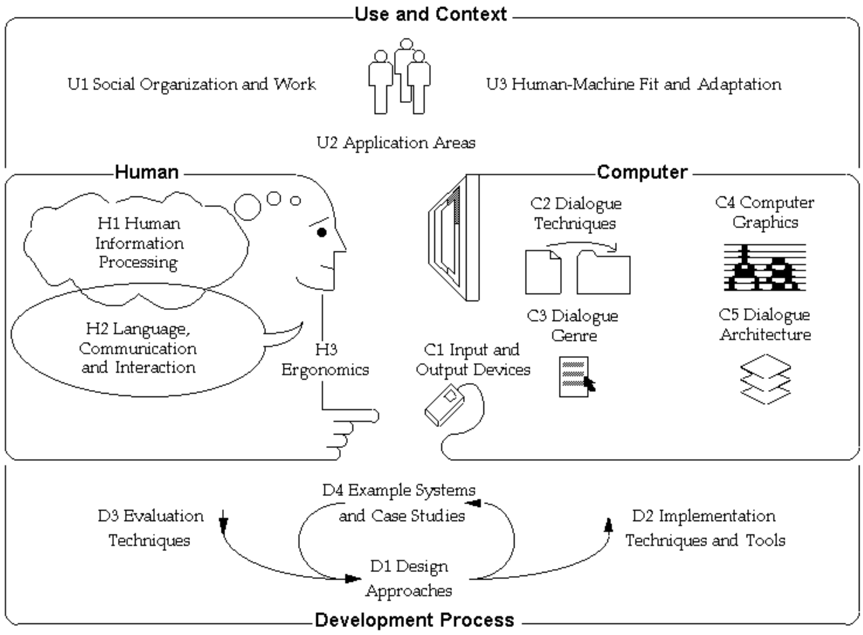



**Figure 1-2: Human-Computer Interaction as described by the SIGCHI Committee (Hewett and others, 2002).**

As can be seen in the illustration the problem domain is multi-faceted, even while examining the problem only from a Human-Computer Interaction perspective. The illustration also shows the overlap with other research areas relevant to the research presented here.

## Knowledge Technologies

Knowledge Technology concerns techniques and methods from the field of knowledge-based systems. Well-known examples are expert systems and decision support systems. Knowledge Technologies represents a new way of organising diverse research topics under one heading. According to a recent workshop on knowledge technology (Kirsch, 2002) the "objectives of the work on knowledge technology will be to provide automated solutions for creating and organising virtual knowledge spaces (e.g. collective memories) so as to stimulate new content and media services and applications." This means that knowledge technologies are situated somewhere between digital content (of all types and sources), and a wide range of applications.

The research in knowledge technologies covers the issues of adding explicit semantics to content (e.g. metadata and indexing, ontologies) as well as acting upon semantic descriptions (e.g. semantics based navigation, device independent interfacing, visualizing knowledge).

An important initiative in the area of Knowledge Technologies is the Semantic Web project of the Web consortium (W3C[18]). The project's goal is to add a layer of machine-understandable data to the Web. The reason for doing so is the believe that the only way the Web can reach its full potential is if it becomes a place where data can be shared and processed by automated tools as well as by people. This project is a successor to W3C's metadata project[19], this shows the roots of the Semantic Web project in metadata research.

This extra layer on top of the existing content of the Web, or better Internet, is of importance for navigating services as well. Such a layer can help users find the service they are looking for.

## Ubiquitous/pervasive computing

The research area of ubiquitous computing was first introduced by Mark Weiser in his defining paper 'The Computer for the 21st Century' (Weiser, 1991). A good working definition of ubiquitous computing is "an environment saturated with computing and communication capability, gracefully integrated with human users".

Ubiquitous and pervasive computing are two names for what is largely the same research area (Husemann, 2001). The difference is that pervasive computing focuses on devices that can be used to access our information or services, while ubiquitous computing focuses on avoiding users having to use computers at all. In ubiquitous computing the focus is on computation in the user's environment to assist task fulfilment without interruption. While this distinction is not clearly made by all researchers their research is still relevant here. Thus in the interest of avoiding confusion and not neglecting relevant research I will use the term 'ubiquitous/pervasive computing' for the field. Fortunately the working definition covers both bases and is a good basis to use here.

Both pervasive and ubiquitous computing address relevant issues when trying to make it as easy as possible for users to find and use services. More access devices might enable users to find and use services in a wider range of situations. Besides, an environment that is augmented with computational devices could take over some tasks from the users. This requires knowledge provided by the research in ubiquitous/pervasive computing.

---

[18] http://www.w3.org/2001/sw/

[19] http://www.w3.org/Metadata/

## Distributed computing

In (Tanenbaum and Steen, 2002) Tanenbaum & van Steen give a clear definition of a distributed system:

> *A distributed system is a collection of independent computers that appears to its users as a single coherent system.*

This definition has two aspects, one of which deals with hardware, and one of which deals with software. The hardware aspect is that the computers are autonomous, while the software aspect is that the user believes that she works with only one system. Although the direct connection between distributed computing and the motivation behind this research might not be immediately obvious, research in distributed computing is a fundamental part of this thesis. Knowledge about distributed computing ensures that the technology is in place to deliver services to the user over time and space. For instance it enables the user to access a service using different devices. It also enables the creation of services that appear to be one but actually consist of multiple services distributed over the network. Research in distributed computing enables most of the technological requirements needed for navigating distributed services.

## The research areas combined

The defining trademark of this thesis is that it combines specific knowledge from each of the areas mentioned above to solve a new and emerging problem. To an extent this is disadvantageous in that it requires a balance between each of the areas. The discussion shouldn't get lost in details of such matters as usability aspects of the use of colour in service interfaces, yet it does need to combine the relevant issues into a coherent research topic.

Not all topics of each research area are immediately relevant. The limitations of the scope coupled with the limitation of available time of this thesis means that only certain topics will be addressed. This is not to say that the other topics are irrelevant in the long run, likely not, but that they are not directly needed to test the hypotheses posed in this thesis. The diagram below gives a clear overview of which topics within the different fields *are* relevant:

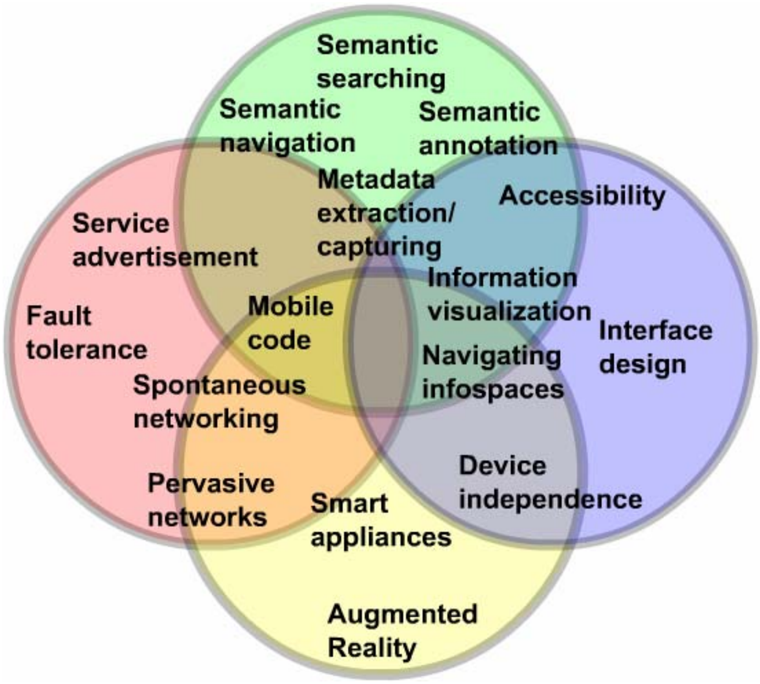



**Figure 1-3: The relevant research topics. Closer to the centre means more relevant.**

Many research institutes are working on the separate topics. However only a few are focussing on the same combination of topics as this study does. Those are the ones that have the same set of problems for solving. It could be said that their starting point is the same scenario as that given above. The table below gives a condensed overview of most of the relevant institutes and their projects. It also shows their goals to illustrate how they differ not only from each other but also from this thesis. I should add one caveat here; the list is not and cannot be exhaustive as new projects constantly come into being and existing ones reach completion.

| Institute | Project | Goal |
|---|---|---|
| **SICS** | sView | Create a framework for user-interaction with electronic services |
| | GeoNotes | Annotate physical places with virtual notes |
| **SICS/ University of Nottingham** | Accord | Construct, administer and manage future interactive home environments |

| | | |
|---|---|---|
| **University of Maryland** | Lifelines | A general visualization environment for personal histories |
| **Carnegie Mellon** | Aura | Provide each user with an invisible halo of computing and information services that persists regardless of location. |
| **University of Washington** | Portolano | Invisible computing |
| **University Twente/CWI** | UWISH | A user-centred design method for Web-based services |
| **Stockholm University** | FUSE/FEEL | Ubiquitous service environments |
| **University of Karlsruhe** | | Ubiquitous and mobile computing |
| **MIT** | Home of the future | How new technologies, materials, and strategies for design can make possible dynamic, evolving places that respond to the complexities of life. |
| **MIT** | Oxygen | Bring an abundance of computation and communication to users through natural spoken and visual interfaces, making it easy for them to collaborate, access knowledge, and automate repetitive tasks. Thus creating a pervasive, embedded, nomadic and eternal system. |
| **NIST** | Smart Spaces | Address the measurement, standards and interoperability challenges of smart environments |
| **Fraunhofer-Gesellschaft** | Ambiente | Develop work and collaboration environments that respond to the demands of new work practices. |
| **University of: Bristol, Lancaster, London, Sussex, Nottingham, Southampton and Glasgow** | Equator | Integration of physical and digital interaction. |
| **University of: Strathclyde, Dublin, St. Andrews, Joseph Fourier** | Gloss | Investigate the barriers, both user-centred and technical, to the construction of flexible and powerful living and working environments |

| | | |
|---|---|---|
| **University of: Aarhus, Lancaster** | WorkSPACE | Software components and hardware artefacts that may be combined and integrated into augmented reality work places, environments, and fields. |
| **Georgia Institute of Technology** | Future Computing Environments | Investigate, prototype, and construct computing environments that might be commonplace in 10-15 years. |
| **Georgia Institute of Technology** | The aware home | Addressing the fundamental technical, design, and social challenges in creating a home environment that is aware of its occupant's whereabouts and activities. |
| **EU/Philips** | SPATION | Find innovative solutions for the movement, organization and retrieval of information in future home networks consisting of CE equipment, PCs and handheld devices. |
| **University of Nottingham** | MiME | Design systems that better promote human experiences around the collection, storage and sharing of intimate media instead of allowing intimate media to disappear into the computer. |
| **Industry labs** | | |
| **HP** | Cooltown | Pervasive and nomadic computing based on Web technologies |
| **Accenture** | HomeLab | The home as the premier ubiquitous computing and critical service delivery platform |
| **Multiple companies** | Application Service Providers (ASP) | Providers host software applications on their own servers and have customers rent the use of the application and access it over the Internet. |
| **Multiple companies** | Web Services | Inter-application messaging across the Web. |

A further caveat is that while the projects have an overlap of interests that should not be taken to mean that they're focusing upon exactly the same problems. Each deals with the four research areas but their focus can differ. Some may take a technical approach and concentrate upon issues as spontaneous networking or making devices intelligent. Others may take a humanistic approach and focus upon how new technology fits into family life.

## 1.4. Hypotheses

Few of the issues identified above can be the topic of this thesis. Not every issue is unique, 'nor will each issue fit the scope of this research. Some of the questions can be answered by existing theories, while others are simply too big for this project. The issue in simplifying finding and using services that I find particularly interesting is *navigation*.

> *How can we make the task of navigating services as simple as possible?*

A preliminary literature study showed that some of the questions relating to that area have not yet been the subject of much research. Ben Schneiderman stated the need for such research during the initial phase of this research project in an article for the Communications of the ACM[20]:

> *Attaining the benefits of universal access to Web-based and other information, communications, entertainment, and government services will require a more intense commitment to lowering costs, coupled with human-computer interaction research and usability engineering. A starting point for research would be a program that addressed as least the universal usability challenges of technology variety, user diversity, and gaps in user knowledge.*
> *- Ben Schneiderman (Schneiderman, 2000, 85-91)*

The universal usability challenges of technology result particularly from the changes discussed in the introduction and are a fruitful ground for research that potentially extends existing theories. In this regard navigation of services is a promising topic that, to judge by the content of 'Human-Computer Interaction in the new millennium' (Carroll, 2002) is sadly neglected. Carroll's book gives an overview of the direction in which the field of Human-Computer Interaction is currently heading. It would appear that the subject of navigation of Information Space had its share of research interest over the past decades, but that that interest is declining. Nevertheless the changes mentioned in the beginning of this chapter provide new opportunities in this area.

What will be different in the task of navigation as we move from locally stored documents and applications to distributed ones? What if the distributed collection is accompanied by a large number of services? And what if the users want access not only from a desktop computer but from a wide range of devices as well? How would that affect users' navigation behaviour? What does that mean for the entities being navigated and the way in which they are navigated?

---

[20] Association for Computing Machinery.

Those questions will be the overall focus of this research, which brings us to the general research question:

> *How does the transition from an environment existing of local documents and applications to an environment of distributed services and documents affect navigation?*

The goal in answering this question is an initial, innovative, constructive, and informed investigation into the realm of services navigation. The scope of this question is deliberately broad, as its sole purpose is to delimit the area of interest. Within this area of interest I pose the following three hypotheses:

> **Hypothesis 1:**
>
> *There are differences between how users navigate in search of information (documents) vs. how they navigate in search of functionality (services).*
>
> **Hypothesis 2:**
>
> *A geographical metaphor is more useful for services with an 'actual geographical location' than it is for documents.*
>
> **Hypothesis 3:**
>
> *Different metadata is required from services than from documents to implement the different navigational user interfaces.*

This thesis will result in an indication of the validity of the hypotheses as well as a platform for navigating services and accompanying services. Both of which can be used for further studies.

Besides engineered products experimenting with possible solutions to newly arisen issues yields insight into what will be the most fruitful direction for future research and development. Therefore the result will also include an overview of the most promising directions.

The remainder of this thesis serves to show how the hypotheses were identified and how they were tested for validity. In discussing the results I will not only show that the research indicated that the first hypothesis is valid, but also provide specific details about the differences in how users search for information vs. functionality. With regards to the second and third hypothesis I will explain how the research indicated that they hold but only in a limited domain. For that I will provide a detailed overview of the required metadata for services and the differences from the metadata required for documents.

## 1.5. Thesis outline

The next three chapters will provide a more thorough exploration of what the effects of the changes will be and what has, and hasn't, yet been researched. After constructing a number of relevant hypotheses a study will be set up to test them for validity.

For reasons that will be explained in the methodology chapter the research in this thesis is firmly rooted in the *scenario-based* research tradition. Scenarios are a recurring theme throughout this text as they are useable at many stages and can supply a common ground for discussion. When discussing one of his projects John Carroll, founding father of scenario-based research, pointed out how scenarios can be useful in finding problems as well as answers:

> *"Our project was essentially exploratory – we were searching problems and solutions at the same time"*
> *- John Carroll (Carroll, 2000)*

This thesis will first of all use scenarios to clarify the hypotheses. Thereafter scenarios will be used as a guide for developing and implementing a prototype. The prototype in turn will be used to implement a scenario to test the hypotheses for validity.

The research of this thesis is partly theoretical and partly engineering. The results will both be an extension of the theories on navigating Information Spaces with knowledge about navigating services, and a technical exploration resulting in a prototype platform for navigating distributed services.

In the next chapter two scenarios will be worked out and analysed in detail to give an insight into what new questions arise after the three changes mentioned in the beginning of this chapter took place. Chapter 3 will then explore existing theories on navigation in general and navigation of Information Spaces to see whether existing theories are applicable. Subsequently chapter 4 takes the theories of navigating Information Spaces and translates them into the area of navigating services. It includes an exploration of the state-of-the-art in the area of navigating services to see whether someone has already encountered these issues, and, possibly, solved the problems identified thus far. Thereafter the need will arise not only to explicitly state the hypotheses but also to choose an appropriate methodology for testing their validity. A task accomplished in chapter 5. Finally the remaining chapters will describe the implementation and test results followed by a discussion about the implications of the findings.

## 1.6. Summary

This chapter introduced the problem domain of this research. It discussed three current trends in information technology that might change the way we work and play. Trend one is going from local to distributed documents and applications (services). The second trend is going from distributed documents to distributed services. The third trend is going from one to many devices for accessing the distributed documents and services. The global effects of the trends were discussed and a problem domain delimited; the effect on navigation. It included a brief overview of the relevant research areas and topics, and the institutions conducting similar research. Finally the outline of the thesis was given. The next chapter provides a deeper analysis of the effect of the changes.

# 2. Scenarios

*The process of identifying proper hypotheses starts with an overview of the potential effects of the three changes mentioned in the previous chapter. Using scenarios as a vehicle this chapter provides such an overview. Each scenario will be analysed in the light of the research topic and the relevant issues will be highlighted. Subsequently the general research topic will be narrowed down to several more specific topics.*

The reasons why scenarios are a useful tool and form the basis of this research are detailed in the methodology chapter. At this stage however the definition of a scenario may be of use to the reader:

> *Scenarios are partial descriptions of system and environment behaviour arising in restricted situations. They are instrumental to the following activities: describing and clarifying the relevant properties of the application domain, uncovering system requirements, evaluating design alternatives, and validating designs.*
> *- Benner et all (Benner and others, 1993, 117-134).*

The reader will find that the scenarios below will give rise to research questions. Subsequent chapters translate the questions raised by the scenarios into the hypotheses that form the core of this thesis. Once the hypotheses have been identified and the methodology for testing their validity chosen, I will return to the scenarios and use them to both develop and test prototypes. The scenarios therefore will serve first as a guide detailing what the requirements for a prototype are, after which they can be used to set up a test situation.

So that the range of issues upon which this research will focus are fully covered there will be two scenarios, each of which has a slightly different focus. The first scenario primarily serves to clarify the likely effects of the changes introduced in the first chapter. It has a storytelling character[21]. The second scenario focuses upon the topics chosen for this research and will be referred to frequently to, among other things, identify requirements for the prototype and create a use case for it. The second scenario focuses more upon specific issues, is more compact, and uses somewhat less narrative than the first.

[21] See any work by Oliver Sacks for some outstanding examples.

## 2.1. Why scenarios?

John Carroll (Carroll, 2000) points out that real world experiences can be an excellent source for scenarios:

> *The most revelatory scenarios come from opening ourselves to the real world – from noticing what really happens in human activity.*
> *- John M. Carroll (Carroll, 2000)*

I concur and have used my own experiences as the basis for a scenario. It is based upon events that revealed to me current day problems regarding the use of everything that has a "digital heartbeat," and chose to write down the experience that others may read about why I work in this field. The experience inspired me to try and solve some of the problems I encountered, and this thesis is the first fruit of that.

The scenario is perhaps slightly unorthodox; it focuses not upon one specific problem, but rather upon the range of problems relevant to service provision as it now stands. In line with Caroll's guidelines it has been lightly edited so that the larger picture may be kept in focus, and the "storytelling" methodology of scenario-based research is kept intact[22]. It should also be said that while it is rich in detail and raises many interesting questions, only those aspects relevant to this research will be highlighted during the subsequent analysis. Thereafter the scenario will be repeated using a futuristic setting in an attempt to suggest how the situation might have changed when the changes alluded to in the introduction have taken place.

## 2.2. Scenario 1: The trip

**Present**

It was March, and I needed a break, I had been working overtime since Christmas and felt that some quality time with my girlfriend D was in order. She was living in the Netherlands while I was working for a year in Denmark. We agreed upon spending a week somewhere in Denmark. So first thing I did was to browse the Web for information about cottages along the coast. Full of enthusiasm I started searching but even though I'm an experienced Web surfer I somehow just couldn't find the right

[22] Which might explain the colloquial English which, according to Carroll, can be an appropriate style for a scenario (Carroll, 1995).

information. I kept on searching, not least because I'm too stubborn to give up and say, "I can't find it". Alas time defeated me ultimately I had to give up.

Providentially my colleague J knew of a place for us, his parents' cottage "somewhere on an island in Denmark!". I tried to look it up on the Web and after having searched nearly all the wrong places (and a few crashes) I found a map of the town in which the house is located. It sounded great so we decided to do it. It had extra bedrooms so we could even invite some friends from the Netherlands over for a few days.

Since I knew where we were going I could collect some information about the region. I wanted to get all the usual information of use to tourists such as; "where's the nearest McDonalds?" D did likewise. Once again I was amazed at the difficulty in finding the information about this specific place. I could find some information, but not exactly what I was looking for. There seemed to be a lot of redundant information on the Internet, each small town seemed to have twenty Web sites containing tourist information and all of them claimed to "have it all". The reality however was that most of them were last updated just before Christmas 1997. Strangely D sent me the bookmarks[23] that she had made while searching for the same information and they were markedly different. There was some overlap, but taking into consideration the topic, the diversity of Web sites to go was absurdly large. Ultimately we decided to forget about finding information online and to go straight to the tourist office once we arrived at our destination.

J's parents had to give me the key to the house and the rent was payable in advance. J called his mother and arranged to meet that evening at his place and exchange keys. The next day J took the key to work and handed it to me after which I drove to the local ATM on the university campus to get the rent in cash. A firm handshake and a look into each other's eyes and the deal was done.

Thus it was that we were a happy man and his girlfriend looking forward to a relaxing week. For J however there were still some things that

[23] A stored link to a Web page for quick retrieval later.

[24] Which is pretty difficult to be honest! But I digress.

needed arranging. First he had to tell a friend of his parents who lives close to the cottage that we were coming and ask that he open the curtains, check for spiders, turn on the heating, hook up the television and the like. Next J had to give me some special details about the house. For instance, how to turn on the ‘fridge, - there’s a special switch hidden behind the fridge.

Coincidentally I would receive a new laptop computer during my holiday. This meant that I had to turn in my old one on the Friday before I left, and reformat the drive. Luckily I have a PDA (personal digital assistant) to take my personal data with me all the time so I unhesitatingly emptied the hard drive of my old laptop after backing up.

Sunday was the big day. I had agreed with D to meet around one o’clock in the cottage - As she had to drive approximately six hours from the Netherlands we couldn’t make an exact appointment. I picked up some clothes, my PDA, my guitar and jumped in the car and drove the two hours to the cottage. I had to pass a toll bridge but as I had lost my credit card a week before I had to take some cash with me. For this I had to visit an ATM machine. Since my bank is not “up to the latest and greatest” in information exchange I could only use my bankcard with their own ATM machines. They are not “all over town”. That accomplished I went on my way, passed the bridge and went straight on to the cottage.

Arriving at the cottage I was pleasantly surprised by the exterior, it looked very nice, with a big garden and everything I might need. Opening the door on the other hand was like getting off the airplane from Iceland to Kenya. It was HOT. The first thing I did was search for the thermostat while holding my breath. I couldn’t find it and had to run for the garden gasping for fresh air. After several such runs I shut down eight of the ten heaters and the temperature slowly dropped. Half an hour later it was safe to enter the house! During the following week we had lots of trouble with the heating. It transpired that there **was** no central regulating thermostat we had to control it by hand.[24] Anyway, I made myself comfortable and started waiting for my girlfriend.

Since the house had no phone and I had no mobile phone (I was waiting for a PDA to incorporate a phone without growing in size) I was not able to call D, who did have a mobile phone. An hour passed and I grew pretty anxious. Even more when she was two hours late. That was annoying, being known to friends as Mr. Always-Reachable I was suddenly unable to contact others. I took a chance that D would not arrive within the next half hour, grabbed my PDA and drove to the nearest gas station to find a phone. Unfortunately D's mobile phone couldn't be reached so I tried my parents, they were home and had called her earlier that day. They told me that her car had broken down just across the border between Germany and Denmark. And that she was waiting for the road service. There was nothing I could do so I went back to the cottage to wait. An hour later I drove back to the gas station and finally managed to reach her. She told me that she had been waiting for the road service for three hours before they found her, although she had called them repeatedly on her mobile phone to explain to them where she was. And that she had done this until her 'phones batteries were nearly depleted. The whole operation was probably going to take another hour so I went back to the house and waited another two hours but she still didn't arrive. Again I drove to the gas station and called her at the garage. D explained the problem with the car and it turned out that it could not be fixed until the next day, so I had to drive an extra three hundred kilometres that evening to pick her up. At last we met. Later that week we drove back to pick up the other car again.

Finally we both were at the same place and our vacation could start. No more problems from then on until I programmed my PDA to wake me up in the morning in time to let in the heating engineer who was to check the heating. I woke up half an hour late wondering why. I checked my PDA but it had crashed and asked me if I wanted to erase all my data! Remembering the fact that this small device was the only thing I had with me that had my personal data I panicked. Fortunately I was cool enough to press 'no'. Which didn't help. It kept asking me if I wanted to erase all data. It was persistent. There was no other option than resetting the PDA by pressing a paperclip through a tiny hole on the back of the device and erase all data anyway. Now I was completely out of data and had to get back to pen and paper again. To add insult to injury later that week I couldn't call my friends because I no longer had their phone numbers. Great. - The mechanic said the heating system was working fine though.

### 2.2.1. Analysis

Within the scenario the first, and arguably the most important problem addressed is finding the right information. Most readers will be familiar with being unable to find a certain piece of information on the Web while knowing that it must be there. The number of Web pages has risen exponentially since the introduction of the Web, currently some search engines purport to search a total of 2,469,940,685 Web pages[25]. Finding one document in such a massive collection is hard, even with the help of existing search engines.

Tim-Berners Lee calls search engines "notorious in their ineffectiveness" because they have to read Web pages and try to deduce the sort of questions it might answer (Berners-Lee, 1997, 57-58). The problem is not with the search engines but with the fact that it is impossible to extract every possible meaning of a Web page, even with the help of technologies like natural language processing and artificial intelligence. Matthew Chalmers (Chalmers, 1999) also points to the problems of information navigation that is solely based on the *content* of each information object *in isolation*. He starts at the bottom of information representation quoting Gödel's demonstration of the problem of inconsistency in logic and mathematics for information representation (Gödel, 1962)[26]. Wittgenstein (Wittgenstein, 1953) took Gödel's work and replaced the axiomatic basis with a social, linguistic one to reinforce the view that mathematics was part of human history and not independent of it. Together Gödel and Wittgenstein discredited positivism[27] by showing that mathematics is not suitable to "consistently, truthfully and absolutely represent the world". The positivists' concept of naming and reference, which connected one mathematical symbol with something in the real world, turned out to be unjustifiable.

Language suffers the same problem, as Chalmers continues to explain. It underlies mathematics and forms the basis for all our modelling and consideration of the world. It is a *subjective* system attuned to the social constructs in which it exists. But, as

---

[25] www.google.com

[26] Interesting books on the same topic are Douglas Hofstadter's 'Gödel, Escher, Bach' (Hofstadter, 1979) and Roger Penrose's 'The Emperor's New Mind' (Penrose, 1989). They further discuss how Gödel's proof of 'unprovability' for a general arithmetic has broader consequences for humanity. They conclude that it is unreasonable to expect a legal system, a set of physical laws, or a set of religious commandments to have the same rigor that characterizes arithmetic. Nevertheless, one can expect examples of logical statements to arise that are neither provable, nor disprovable, within a complete logical framework.

[27] The system of Auguste Comte designed to supersede theology and metaphysics and depending on a hierarchy of the sciences, beginning with mathematics and culminating in sociology. Positivism intends to find knowledge about the world by empirical means, rather than by relying on dogmatic assumptions based on religious or philosophical authorities or traditions.

Chalmers put it, "there's no escape from language". Even if we try to formalise language we will fall back to the language itself when using these formalisations. This means that there is no escape from the subjective aspect of language. De Saussure, as part of the structuralism paradigm[28], noted this dynamic aspect as well when stating that elements of a language only have meaning because of their relationship with other elements (Saussure, 1959). This contrasts with the positivist view of a one-to-one relation between a language element and something in the real world. So the meaning of a word can change over time or between situations, which undermines the positivist approach. This means that regarding information navigation you can't assume that the words in a document faithfully and fully describe the meaning of it. This is why Tim-Berners Lee has a limited belief in search engines that have to retrieve meaning from just the words in the documents. It also lies at the heart of Nardi & Barreau's (Nardi and Barreau, 1997, 76-78) question as to whether documents in the workspace as a collection can be easily characterised, ordered and retrieved based upon common characteristics or full text retrieval.

The arguments above imply that there is no one-to-one match between one meaning, which users for instance try to capture in words and enter as a query in the search engine, and the actual words in a document. When searching for information users try to match their meaning with the meaning of a document, which requires them to translate their meaning into words that *they think* matches the meaning in their head and match them with words in a document. This is an error prone translation process. This fundamental problem is highlighted in the scenario where I was unable to unambiguously translate what I needed into words and find the right document. The documents returned by the search engine all matched on words but none on meaning. It shows that searching information based solely on the content of a document, which is today's practice, is rather ineffective. The lack of contextual information makes it difficult to find a certain document.

Another complicating fact encountered in the scenario as it pertains to searching is the amount of overlapping information. Since many Web pages contain the same words many match a query consisting of a few words. Searchers are therefore often confronted with many matches requiring them to sift them manually. Even though the returned matches might contain largely the same information, users can never be sure of this so they are inclined to read through more than one match.

The second issue opened up by the scenario is the problem of accessing services. Services within the scenario are offered functionalities such as adjusting the heating in the house, getting money from the bank, messaging, getting access to the house and so

[28] Structuralism is based on the belief that cultural objects such as literature, art, architecture, etc. cannot be understood in isolation, they must be studied within the context of the larger structures to which they contribute and within which they developed.

on. All those services were needed to prepare for the trip. The problem is that a lot of the services involve physical goods or persons to be moved around. For example to get access to the house a key was needed that had to come from another place. It had to go from hand to hand to finally end up in with the right person.

The third problem identified by the scenario is the lack of inter-human communication channels. Many problems depicted in the scenario could have been prevented if there was only a way for the two actors to communicate with each other. Be it by voice or text messaging. But the technology for this was either dysfunctional or plainly lacking.

A fourth problem encountered is that users have to take care of their personal data themselves. They have to manually shift it around to have it accessible where they are. They are administrators of their own data. This is true not only for data but for applications as well. If users want to have the same functionality available from where they are, they have to install the application themselves.

### 2.2.2. Future Trip?

Now that a scenario of a contemporary situation has been discussed it is time to look at what the same scenario might look like in the near future, after the changes mentioned in the introduction chapter took place. Again I must stress that this is only one of many possible future scenarios. History has taught us that it is almost impossible to predict the long-term future of technology and I neither intend nor purport to break with that tradition. That is why the future as depicted in the following scenario is the near future (next five to ten years). The only assumptions made are the ones described in the introduction chapter, they are based upon a limited extrapolation of changes already taking place at the time of this writing. Although the extrapolations are based on careful investigation and likely to take place, one can never be sure, meaning that the results should always be interpreted with care.

*This* version of the scenario is partly technology driven since the three changes from the introduction chapter are technology driven, but it is also user driven. A dichotomy is often assumed between technology-driven and user-driven design, but they are complementary as Carroll hints at.

> *...it behoves us to generate scenarios that emphasise the special, novel affordances of new technologies... [They are] a starting point and reference point for subsequent software development work.*
>
> *- John M. Carroll (Carroll, 2000)*

Although there are many factors determining the success of a technology, a technology-driven scenario will be useful to get to something that can be tested.

**Future**

I had to take a break. I informed my software agent about this a couple of days before so it had been collecting all kinds of relevant information for a weekend trip for me and my girlfriend over the last few days. Based on some statistics and personal information it could deduct enough information to start searching. I told it to look specifically for a cottage in Denmark so it contacted other agents, notably some from companies specializing in weekend trips, and negotiated with them for a fitting trip. It saved me searching the Web by hand; I could just pick one from the list my agent presented me on returning.

Coincidently the agent also found the cottage of my colleague's parents, which looked just fine so I picked that one. I sent my agent on its way with the booking order. No need to think about it anymore, if there's anything the agent still needed to know it could always come back and ask me. Of course I only needed to sign for the payment. Since some friends of mine wanted to come over I gave the agent the assignment to inform them about this cottage as well and ask them for their approval. A few hours later the agent returned after contacting the personal agents of my friends and checking their schedule and asking for approval. I gave it my permission to book the cottage. I also ordered it to collect information about things we could do in the region.

Paying for the cottage meant nothing more than granting my agent the right to handle the payment. I just assume that everything will go smoothly. No running to the bank, handing over money or receiving receipts, just granting authorisation to my agent. After a while the agent returned confirming the booking. I called my girlfriend D in the Netherlands saying I booked a small house and I was looking forward to a nice relaxing break with her there.

I informed my colleague J that I booked his parents cottage for next week and he started to make a few preparations for the house. First thing he did was accessing the network of devices running at the cottage to check the status of the house. Since the network didn't send any messages during the last weeks regarding problems, he expected everything to be just fine. He turned on the Web cams installed in different places around the house, he checked the temperature, turned on the heating and the fridge, checked the lights, gave my personal identification pass permission to enter the house, uploaded a few movies to the home gateway server, set the clock to summertime and made sure my PDA would be able to access the network so I could get to all the services in the house using my PDA.

The manufacturer is monitoring many devices in the house. The fridge for instance notifies the manufacturer the moment something is wrong so they can fix it. If they're not able to fix it remotely the fridge sends out a message to the owners with an understandable description of the error so they can decide what to do.

There are also software services running on the home network. The owners have subscribed to a number of services that they think interesting. For example a service that gives frequently updated information about things to do near the cottage. A program guide to show what will be on television that night, and several similar services. Service providers maintain both. The only thing the consumers have to do is subscribe after which the provider has the obligation to keep it running at the consumers' home gateway server.

Although I swapped laptops just before my short vacation, it didn't take any unnecessary time. My personal data is stored on the Internet so it is always accessible. I didn't have to worry about backing it up from my old laptop and restoring it on my new one. The software I use is network-distributed services to which I subscribe, which means that I pay per use and only download the service when I use it. This saved me the bother of reinstalling software on my new laptop.

The day I drove to the cottage I had little to worry about. All relevant messages were available through my messaging service, which was accessible from my car display. In case I was driving somewhere where there was no wireless network available the messages would be cashed in a database in the car, so I had all my data available while driving. I had my mails and favourite Web sites read out to me through the loudspeakers during the long drive. Underway different services popped up on my car system giving me all kinds of info relevant to the area I was driving through. By specifying I was hungry I found a nice restaurant in the area where I could buy a small meal. I could pay the toll bridge with my mobile phone; the fee was added to my telephone bill. I was on my way.

D drove from the Netherlands to cottage in Denmark but was less fortunate. Normally her car is running an internal network connected to the Internet, which allows vendors to monitor its different parts, but one of the tubes for the cooling started leaking and that part was not being monitored. This was at the border between Germany and Denmark and she had to pull over. Luckily the onboard network noticed there was something wrong because the temperature of the engine rose and it already collected the relevant information to deal with the situation. It showed the telephone number for the road help on the dashboard display and described what the problem could be. It also gave information on the nearest garages so D could decide which one to contact. With it the network offered to send the garage the info on the error so they could anticipate on that. She didn't have to make one phone call if she wanted to. Since she was standing at the verge of a very busy highway she had to get out of the car and get away from the road. She wanted to call the garage herself so she took her PDA. Outside the car she wirelessly connected it to the network in the car and had access to the data again. She phoned the garage and two hours later help arrived.

In the meanwhile D contacted me through my agent since she didn't know my exact location. She sent me a message saying she would be delayed due to problems with her car. I was a little surprised and sent back a message using the voice interface of the onboard messaging service. Fortunately she assured me things weren't that bad and she expected to arrive around two hours later.

Arriving at the home I identified myself with my personal identification pass and got in. To my surprise it was very hot inside so I turned on my PDA, connected to the local network running in the house and searched for the heating. I found it pretty fast and turned off eight out of the ten heaters. Slowly the temperature dropped.

There was nothing I had to do but reading the news that D was on her way again and expected to arrive within 3 hours. Once she arrived we watched some television and went to bed. I wanted to set the alarm on my PDA but it crashed. There was no way I could restore the data on it. Luckily my data was stored online and I could access using other devices. As far as an alarm clock goes I used D's PDA and search the Internet for an alarm clock service. I found one, which could be used by the PDA and made a micro payment and set it for the next morning. We got up on time.

### 2.2.3. Analysis

The tasks in the scenario are the same as in the 'present' version but technology makes it a different story. The most striking difference is the central role of the network (the Internet) in the 'future' scenario. Whereas in the 'present' version the Internet mainly served as a huge database of documents, in the 'future' version it offers functionality (services[29]) as well, thereby fulfilling a larger part of the user needs. It offers interpersonal communication, control of devices, banking services etcetera. The tables below give an overview in list form of the devices and services used in the scenario.

| Device | Function |
| --- | --- |
| Desktop computer | Browsing |
| Web cam | Taking photos |
| Heating | Heat the house |
| Fridge | Refrigerate |
| Lights | Turn the light on |
| Clock | Tell time |
| Home server | Gateway/storage |
| PDA | Accessing and controlling the house |
| TV | Display |
| Laptop | Workstation |
| Car display | Display |
| Loudspeakers | Generating sound |
| Mobile phone | Voice messaging |
| ID pass | Identification |

**Table 1: A list of devices used in the scenario and their functions**

[29] As defined in chapter 1.

| Service | Description |
|---|---|
| Agent[30] | Finding information<br>Combine information from different sources (calendars of user and bookings office)<br>Inform friends<br>Negotiate with other parties/agents |
| Instant Messaging | Contact and chat with others |
| Web cam | See images from the Web cam |
| Heating | Control the heating |
| Fridge | Control the fridge |
| Light | Control the lights |
| Clock | Tell the time |
| Storage | Store data |
| Access | Specify access rights to the cottage |
| Display | Display visuals |
| Sound | Make sound |
| Voice messaging | Send voice messages |
| Identification | Check access rights |

**Table 2: A list of services used in the scenario**

What follows is a list of relevant issues the scenario gives rise to.

## Technology

The most obvious question is probably whether the scenario is technically feasible. Many assumptions are made on the technology side to construct the scenario. Such

[30] The term 'agent' is used rather heavily in the scenario and a warning is in place not to confuse an agent with the human-like robots of science fiction. The latter are more fiction than science. As used in this thesis an agent is a software routine that performs an action when a specified event occurs. This can be quite complex, but it is nowhere near human intelligent behavior. For a better understanding of what is and what isn't possible with artificial intelligence I recommend reading (Russel and Norvig, 1994). The Agent Web at the University of Maryland (http://agents.umbc.edu/) is a good starting point for reading about the status quo in this area.

assumptions are based upon a limited extrapolation of the three trends we saw in the introduction chapter. These trends are well documented and form the basis for this research. The scenario differs from the trends with regard to the level of detail. Where the trends were described in general terms, the scenario fills them out with concrete services and devices. While there is little doubt whether the trends will continue, there is some uncertainty whether these specific services and devices will become a reality. They are part of short-term developments compared to the trends and harder to predict. That is why the focus should be on the task when reading the scenario, instead of on the specific devices and services. Controlling the lighting in the cottage for instance could be done by a PDA or a mobile phone, that only partly matters, the main point is the task of controlling the environment with a wireless device.

As far as service provisioning goes there are some issues as well. Probably the most important is that there should be an infrastructure for providing the services that meets the following requirements:

- ***Open***
  It should be open for existing services technologies and infrastructures.
- ***Extensible***
  Allow for dynamic changes to the technology.
- ***Accessible***
  Infrastructure/services should be accessible from within many situations.
- ***Adaptable***
  Infrastructure/services should adept to different devices.
- ***Continuous***
  Services should maintain their session state between devices.
- ***Administration free***
  The user shouldn't have to maintain the system.

(This list is partly based on (Bylund and Waern, 2001))

A system meeting these requirements is indispensable for the scenario to become reality. Without such an infrastructure for services it is impossible to reach a level of usability that is a significant improvement on the current situation.

## Navigation

Finding what you are looking for remains a problem in the 'future' scenario, whether it is information or a service you're looking for. The analyses of the 'present' scenario showed that navigating documents could be a difficult task for users. Add a number of services to the collection of documents and the problems become apparent. An example from the scenarios: in the 'present' scenario the actor started a communication application on a standalone desktop computer to contact his girlfriend.

In the ‘future’ scenario he didn’t have any applications installed on his computer so he used a service from the Internet instead. While in the ‘present’ scenario finding the right application might mean searching through a small number of applications installed on a standalone desktop computer, in the future it might mean searching through a large number of services from the Internet.

The ‘future’ scenario shows that the introduction of a (increasing) number of services makes the task of navigation different from navigating in the ‘present’ scenario. The increase in number of services is due to the fact that the Internet can be used for more tasks. The question of *how* the actors actually found the service they were looking for was not addressed in the scenario, it only showed *that* they were looking for it. This evokes the question whether there is a *difference* between navigating services and navigating documents, and, if so, what this difference might be. Only an answer to that question can give an insight into the question of *how* the actors might have navigated.

If there were a difference between navigating documents and navigating services it would be useful to investigate whether there are new opportunities (and threats) for navigation. It might be that the difference allows for different ways of navigating, such as 3D interfaces or time-based navigation. Or maybe different metaphors can be used for navigating services, for instance ones that weren’t successful when used for navigating information.

## Metadata

Finding a service presents some of the same problems encountered when locating a document. The analysis of the ‘present’ scenario made clear that it could be problematic to extract the meaning of a document solely based on its content. Contextual information is paramount to retrieving the meaning, but such information is largely missing for documents sitting on anonymous Web servers. Furthermore search engines present users with thousands of other documents that have similar or partially identical content. Contextual information as well as information about the content of the document can be added as metadata (Cole, 2002), and help the user or software extract the meaning of the document. It will help, but how effective the metadata language is in describing the document is still topic of much debate, as discussed in the previous section (2.2.1).

The need for metadata didn’t disappear in the ‘future’ scenario. In the ‘present’ scenario users had to search for information *themselves*, which could be a tedious job as the scenario indicated. In the ‘future’ scenario this problem was partly solved by delegating the search to a third party, an agent in this case. Maybe the agent will return more quickly and with better results, but it needs contextual information and/or metadata just as much as humans do. Agents might take some weight of the shoulders of users, but they won’t necessarily provide better answers since they encounter the same limitations in extracting the meaning of a document.

Some of the same problems arise with services. What information, or metadata, about a service is needed for users, human or not, to find it and then judge if it is what they are looking for? How can a service make that information available? And is context just as important?

## Personalisation

Since using services requires more interaction than reading documents it is likely that users will have higher requirements regarding the personalisation[31] of a service. Each user has his or her preferred way of working and preferably a service should adjust to that. This implies that the service should sometimes allow the user to adjust its interface, other times it implies that the software should self adjust based upon information it holds about the user. In short services will have to take into consideration the user's habits and preferences. The issues then become how can users communicate their preferences to a service? And how can that service then store them? Could storing such information together with information about their habits do this? Unlike applications installed on standalone desktop computers preferences cannot be stored on the local computer and retrieved the moment the user logs on again because that would require the information to be stored on all devices the user might use to access the service. Somehow the service has to recognise the user and retrieve the right preferences from a remote location.

There are technical issues as well as Human-Computer Interaction issues involved here. The technical issue is how to create services that allow for personalisation. The Human-Computer Interaction issue is how to provide personalisation while minimising interruptions of the user experience.

## Privacy

A side effect of an increase in use of the network for day-to-day activities is that users are more vulnerable to breaches of their privacy. The 'future' scenario showed that it is likely that most documents and services will come across the network. Unfortunately that situation is more open to spying eyes than working locally. Since services afford more interaction with users chances are that the user will give away more information than with documents. Usually users just read a Web page without returning much information so there's not much to spy upon[32]. So what needs to be worked out is a trust mechanism between providers and consumers of services. But what form should that take? A central registry of trusted partners for a user? A per-use confirmation of trustworthiness? What if a service is actually an aggregation of other

[31] Custom tailoring the information or service to the individual.

[32] It is amazing though what information companies can extract from our current use of the Web (McCandless, 1997)

services, how can the user in that case easily specify privileges to the component services of that aggregation?

## Security

Security is a manifestly important issue and is probably the largest hurdle to be faced by service providers. An example from the scenario that indicates the importance of security is the service for adjusting the thermostat in the cottage; unprivileged visitors should be prevented from changing the heating in the house. Although turning the heat up is rather innocuous, one could hardly imagine the risk when medical equipment becomes available as services on the Internet. The five main aspects of security that have to be taken into account when building distributed systems are:

- Authentication
- Authorization
- Accountability
- Confidentiality
- Integrity

There are technical as well as social aspects to security. From a technical perspective security influences everything from the physical level to the services level[33]. If there is a security breach at any of these levels the whole stack becomes useless. Moreover as may be deduced from the regular news stories about security flaws in widely used software packages that connect to the Internet[34], it is difficult, if not impossible, to make something one hundred percent secure.

The social aspect of security is that users have to *believe* something is secure before they will use it. Even if something *is* secure, the user will be reluctant to use it if she doesn't feel it to be so. This means that besides creating secure services the service providers will have to gain trust with the users.

While security has always been an important issue on the Internet it is clear that when users become increasingly dependent on the Internet security will become even more important[35]. Accordingly the introduction of services introduces a higher rank of user

---

[33] See the simplified version of an OSI model in the appendix or read (Henshall and Shaw, 1988) for a more detailed description.

[34] See http://www.internetsecuritynews.com/

[35] At the exact time of this writing (even) Microsoft realized the importance of this and Bill Gates, the company's chief software engineer, called for a fundamental shift in Microsoft's product development to put consumers' security and privacy concerns ahead of adding new features to the company's products (Wired, 2002). This is a break with previous strategies.

requirements regarding security. Especially at the services level this will require a lot of extra work. The underlying layers already have had the security focus since they have been used for some time already[36].

## Administration

Maintaining the services will be a real problem when their number increases significantly, as was the case in the scenario. A recent article in the Scientific American (Gibbs, 2002) addressed the maintenance problem when discussing a new research area called 'autonomic computing', which is a systematic view of computing modelled after self-regulating biological systems. The goal of autonomic computing is to attack three problem areas that emerge in complex computer systems: cost, availability and user experience. First of all it's becoming too expensive to maintain such systems. There is already a shortage of skilled professionals to maintain them, and it this shortage will only get worse when taking into account the fact that the number of computer devices is forecast to rise at a compound rate of 38 percent a year (Gibbs, 2002). Furthermore most of these devices will be connected to one another and to the Internet. There are not enough experts to maintain them, and humans are too expensive to do this job. The second problem is availability. Complex computer systems are increasingly created out of highly interdependent components that are more and more distributed over a network. Each part should handle the failure of another part of the system gracefully, minimizing the total system downtime. The third problem is the user experience. Requiring users to act as system-administrators is not going to improve their experience with the system since they are not so much interested in the system itself than in what they can use it for. The three problems identified by autonomic computing are relevant in the 'future' scenario as well, and only conscious decisions can prevent the problems from occurring.

Concluding the first scenario I can say that the 'present' version had different problems than the 'future' version, but that some problems overlapped. For example adding proper metadata to documents and services seems to be problematic in both situations. Furthermore some old problems disappear and new ones arise between the two scenarios. One of the new problems is navigating services, meaning finding the services you are looking for. The next scenario will focus on that problem in particular.

---

[36] This is not the place to go deeply in all aspects of security but to get a good overview of the relevant issues I recommend reading Cleassens et al's '*A Tangled World Wide Web of Security Issues*' (Claessens, Preneel, and Vandewalle, 2002). They are one of the first to address the security issues of using services in their excellent article. It's a worthwhile read since they rightfully treat security as both a technical and a social problem.

## 2.3. Scenario 2: Travelling photographs

*"…the practical problem is less one of finding scenarios at all and more one of generating and identifying good scenarios […], where 'good' means scenarios that raise and illuminate key issues of usability and usefulness, or that suggest and provoke new design ideas"*

*- Carroll, p.255 (Carroll, 2000)*

In this section a new scenario will be worked out followed by an analysis of the issues it raises. It is worked out in threefold: for the past, the present and a future situation. The reason for this is to point out the old problems that were solved and the new ones introduced with each transition. This in turn will highlight the changed requirements for each situation. The future version of the scenario is the most elaborate one since that is where the focus of this thesis is. One should keep in mind that the future version of this scenario is also based on the three assumptions made in the previous chapter. They are assumptions and therefore the same caution with interpretation as with the previous scenario is necessary.

As mentioned in the introduction of this chapter this scenario is more focused on a specific research topic. It is therefore more compact and less story-like. The previous scenario highlighted most of the issues and it would be redundant to repeat them here. Instead this scenario will be analysed with special attention to one particular issue; that of navigating services. Once there is a clear understanding of the issues regarding navigation we can proceed to the next chapters to see how this fits with existing research, and set up an experiment to answer some of the open questions.

The scenario is about a girl who travels Europe and visits her family in Romania. There are three tasks she tries to accomplish during this trip. First of all she wants to give a slideshow of all the photographs she has taken so far. Secondly she wants to give relatives in Romania hardcopies of some of the photographs, and lastly she wants to update her travel journal for her family at home in the Netherlands.

**Past**

Alana was touring Eastern Europe for two weeks and planned to visit some relatives living in Romania. Before she left she bought a camera and brought it together with a dozen films. Carrying a rather heavy camera with her all the time during the summer in Romania was something she could have done without she later realised. When she arrived at her grandmothers house during the second week she took lots of photographs for her photo album. Her grandmother asked if it was possible for her to get a copy of the photographs, Alana replied that she would send her a copy when she got home. While saying this she remembered she had to take some photographs for the travel journal she was writing for her parents at home in the Netherlands. She sends them in instalments so they could read and see her adventures while she was still travelling. For this she had to print the photographs at a local shop around the corner. Fortunately they could develop the film roll within four hours.

After dinner Alana went to pick up the photographs. On returning she discovered that the quality of the photographs was not as good as she expected, but good enough to gather around the kitchen table with her relatives to give a small slideshow.

Tired from the dinner she went to her bedroom to work on her travel journal and go to sleep. She picked out the best photographs, glued them onto a sheet of paper and wrote some comments below them. The next morning she asked her grandmother to post the instalment of the journal for her. After saying goodbye to all her relatives she left.

Upon returning home Alana immediately had the rest of her film rolls developed and sent the requested photographs to her grandmother.

**Present**

Alana was touring Eastern Europe for two weeks and planned to visit some relatives living in Romania. Before she left she bought a digital camera and brought it together with her laptop to make sure she could take as many photographs as she wanted. When she arrived at her grandmothers house during the second week she made lots of photographs. Alana wanted to put some of these photographs in her online travel journal for her parents at home to see, and print a few for her grandmother.

Step one was to get the photographs from the camera onto the computer connected to the printer. This turned out to be impossible since she didn’t bring the driver for the camera and she couldn’t download one from the Internet. So she had to download the photographs onto the laptop and from there send them to the computer with the printer (to her Web site). Connecting the printer directly to the laptop could have been another option if the printer only had an English instead of a Romanian driver. She took her laptop, connected it to the camera and turned both on. The preinstalled software recognised the camera and started an application to download the photographs. When finished she gave a quick slideshow. Some photographs were too dark so she started a photograph-editing application and adjusted their brightness.

The computer with the printer didn’t have a network connection at all so the only way to get the photographs over was by using a floppy disk. Alana’s nephew started the computer connected to the printer, copied the photographs from the floppy disk and started an imaging application to print the photographs.

Step two was to put some photographs in her online travel journal, but her grandmother didn’t have an Internet connection. Fortunately she brought a GSM mobile phone that could transmit data. She connected the phone to the laptop and after a reboot the phone showed up in her connection panel on the laptop. Now all she had to do was use that connection to hook up to the Internet. Now the phone could be used as a regular modem. The nephew called the Internet provider and read out the needed network settings. Alana configured her laptop and successfully connected to the Internet. She started an FTP application and made a secure connection to the Web server hosting her Web site. She downloaded the correct HTML page, inserted the photographs and uploaded the page back to the server again.

**Future**

About a decade later Alana was again visiting her relatives in Romania and took some photographs with her new digital camera. Alana's grandmother's eyes were not that good anymore and she had troubles looking at the small screen of the camera to see the photographs. Alana noticed this and suggested to watch the photographs on the wide-screen television. She switched the camera to networking mode to search the local wireless network for displaying services. A large number of services immediately "popped up" on the text-based screen. After using a filter the list only showed displaying services, being the television, her laptop and two computer monitors. With a stylus she clicked the text 'television' and up popped the text-based user interface for the displaying service offered by the television. But the camera screen was too small for navigating around so she turned to her laptop. It started with a service browser[37] scanning the local network for services. The default interface of the service browser was a 2D representation of her surroundings, so she could see the floor plan of her grandmothers house with all available devices shown as icons on top of it. She switched to a 3D representation and navigated to the virtual version of the room where she found her digital camera and the television. Next she had the option to start with the photographs and search for a displaying service, or the other way around. Alana chose the former. Clicking the 3D icon of the camera opened a 2D user interface for the camera. From there she could select a photograph and open the services-menu to see which services could handle photographs. She picked the displaying service offered by the television. This required authentication so Alana asked her grandmother to fill out a username and password and the photograph showed up on the television.

Alana suddenly remembered there was sound with the photographs. Using the laptop she opened the interface for the displaying service (TV) again. It could search for services to cooperate with. She found a sound-playing service and checked the service information to see whether it were the speakers in the room, which they were so she turned them on. A volume control automatically appeared in the interface of the displaying service.

[37] A service browser is a program that let's the user look through a set of services.

Alana also noticed some photographs being too dark. She could have used the television itself to edit the photographs but she wasn't sure how realistic its colours were. Besides that she wanted to sit comfortably behind a desk for the job. Using her laptop she opened the interface for the camera and opened the services-menu for a photograph that needed to be retouched[38]. The service browser kept track of Alana's preferred services and one of them was a photograph editing service. The service browser, using Alana's digital passport, identified her as an authorised user and the service showed up.

Next Alana wanted to print some of the photographs for her grandmother. Fortunately her grandmother recently bought a standalone network enabled printer that provided a printing-service by itself. Inspecting the service menu of a photograph for the second time Alana found a printing service, but she wasn't sure it was the right one. To be sure she switched to the 3D interface of the service browser and navigated to the printer. She double-clicked the 3D icon and a 2D interface showed up from where she could find the photographs on her laptop in the usual way. She adjusted the settings and started printing.

Finally Alana wanted to update her travel journal from the garden using her grandmother's Web tablets. Using the service browser on the tablet she opened the service for editing her Web site from her collection of preferred services. The service automatically stored pages on her own Web server so she didn't have to worry about uploading or authentication. Alana connected to the camera that was still inside the house, to insert the latest photographs in the page. She added some comments and closed the service again.

[38] Retouching here means improving a photograph by adding details or removing flaws by using an image-editing tool.

### 2.3.1. Analysis

Possibly the most striking difference between the three versions of the scenario is that the actions differed widely although the tasks were ultimately the same; Alana wanted to give a slideshow, print a few photographs, retouch some photographs and update her travel journal. As the chronologically ordered versions of the scenario show the evolution of technology introduce new possibilities and solve old problems. But the new possibilities introduce *new* problems as well.

As with the 'The Trip' scenario the main source for new problems is the introduction of services that are distributed across a network. The future version of the scenario introduces network-enabled devices offering their services *via* the network. Furthermore functionality that was previously only available as an application installed on a standalone desktop computer is in the future scenario available *via* the network as a service, for example the displaying service, the sound service, the image-editing service and the storage service.

Compared with the past and present versions of the scenario the future version solves many of the old problems and introduces new ones. An example is the displaying service. In the past Alana had just one way to show the photographs to her grandmother, by having the roll of film developed and sending her a copy of the developed photograph. The problem was that her grandmother had to wait for the photographs to arrive. In the future situation Alana can use a displaying service, provided by the different displays around her grandmother's house. This means the waiting for her grandmother is over but new problems, such as the technical problems of spontaneous networking between the different devices, camera and television in this case, arise. Depending on how far this can be automated this means an extra task for Alana.

Most of the issues have already been identified during the analysis of the previous scenario though, and one of them was chosen to serve as the focus point of *this* scenario, navigation. The interconnectedness of the devices and their services brings with it the problem of finding what you are looking for. Once the devices can 'see' and offer services to each other as well as to the user, the question arises how the user can find a particular service. Each of the tasks Alana performs poses new questions regarding navigation. What follows is a discussion about the issues regarding navigation each of the tasks raises.

#### **<u>Give slideshow</u>**

The first thing Alana wanted to do was give her relatives a slideshow of the photographs on her digital camera. This meant that two devices had to work together spontaneously - the digital camera and the television. Initially she had three navigational options. The first option was to start with the television (service) and find content for it to display. A second option was to start with the camera (service) and finding a display to show the photographs. The third option Alana had was to launch a service browser[39] on another device (for example a desktop computer) and use it to find the camera as well as the television and make them work together. Which option is preferred by different users is hard to say, but Alana chose to start with the camera. Using the camera is faster since it doesn't require Alana to launch an external service browser, even though a service browser might offer a user-friendlier interface since it can be launched on a system that has for instance better input capabilities and a large display. Compared with the limited interaction capabilities of the camera (for example a small screen, small buttons and no sound) this has its advantages.

Regarding navigation there is a difference between using the camera to find a display, or launch a service browser and use it to find the camera as well as the display. Using a service browser requires Alana to find at least one service, the one provided by the camera. This means searching and its accompanying issues. Once Alana found the camera service she might have had the option to easily find displaying services from *within* the interface of the camera service. It sounds reasonable for camera manufacturers to equip their product with a service that can find and cooperate with displaying services. In case Alana uses the display on the camera she has immediate access to the camera service so she doesn't have to search for it[40]. As mentioned before, from within the camera service she probably has an option to find and use displaying services.

**<u>Retouch photographs</u>**

Next Alana wanted to retouch some of her photographs. This task is different because it requires a device that is local to Alana to work with a software service that resides on the Internet. The navigational problem involved is finding the software service and using it to edit the photographs stored on the camera. In the scenario the service browser keeps track of Alana's favourite services, like bookmarks in a Web browser. It would have been more difficult if the photograph editing service had not been part of her preferred services. In that case she would have had to search the Internet for the desired service.

Another issue this task points out is that there is a difference between starting with a service and finding some data (photographs) for it to act upon, or starting with data and find a service that can perform the desired action upon it. Which is better depends on the situation and user preferences. Sometimes only one way is possible, at other times both are possible but the user prefers one to the other. Each way influences the navigation method as well.

---

[39] A service browser is a program that let's users look through a set of services. It may not always necessary to use a service browser; services might be accessible from within other services. The service browser is discussed more thoroughly later on in this section.

[40] The camera service is running on the camera itself.

## Print photographs

From a navigational point of view the print-task in the scenario is essentially the same as the slideshow-task, it requires Alana to make two devices work together, the only difference being that a printer is the output device. What makes the *situation* different is that the printer doesn't have much interaction capabilities besides turning it on or off. This means Alana hasn't got the option to start with the printer and find content to print. The printer doesn't have a display or speech interface needed for interacting with services or the service browser. She has to start with the camera and find a printing service or start with a service browser and access both the camera (service) and the printer (service) from there.

## Update travel log

Data storage and retrieval is the important aspect highlighted by the task of updating the travel log. In the scenario this process is automated, data is stored automatically and the log is updated automatically. In a desktop computer centric world Alana would have had to choose 'save' in a menu structure and navigate to a folder on a hard disk to store her travel log. A similar process would have been needed to open an existing log. In the scenario Alana just launches the travel log service, which starts up with her log automatically. She no longer has to navigate through documents on her personal computer to find her log; her data (which might come across the network) is presented by the service.

## Common themes

Each of the four tasks touches upon slightly different aspects of navigating services but there are some recurring themes as well.

Foremost the scenario hints that navigating services differs from navigating documents. This can be deduced from the fact that in the scenario services are accessed using different devices than might be used for accessing documents. Although slowly but surely more devices offering Web access (the most popular way of navigating distributed documents) are introduced, accessing services can potentially be useful from still different devices. The reason that a wider range of devices can be useful for accessing services is twofold. First a service that is running on a device might offer access to other services, besides access to the service that belongs to the device itself. Using the service on a device to find other services leaves out the need to use a general service browser on, say, a desktop computer. Navigating services can thus be done from more devices than just the ones that can run the general service browser. The scenario gives an example where the digital camera was able to search the local environment for services that can show its photographs. In that situation the camera was the device used for navigating the services. The second reason a wider range of devices could be useful is that services might be useful in *different* situations than documents are. Users can do different things with services meaning that services are useful in different situations. In turn each situation will have its requirements regarding the devices that can be used. In the scenario for instance Alana goes from watching a slideshow to retouching some photographs to writing in her log. Clearly the inter-task variation is greater than with switching between documents. As the situation changes, the 'best fit' device might change as well. An example would be the difference between using a service while driving and not having your hands available for navigating, and using it when sitting behind a laptop computer.

Accessing services from within different *situations* affects the way the user navigates as well. An example of the *situation* influencing navigation is that Alana could choose between a camera and a laptop for finding the right service, in different situations these devices might have been inappropriate or simply unavailable. The situation might also influence which services are available or appropriate. Things could have been different for instance if Alana was at work instead of at her grandmother's house. In that case her collection of preferred services might have consisted largely of those services connected to her work.

The *device* can influence the navigational task as well. This is shown for instance when Alana chose to use her laptop computer for navigating since she found that to be more comfortable. The display of the digital camera only had a small screen, making it harder to navigate. Different devices have different interaction capabilities, which in turn affect the navigational process.

Furthermore navigation is influenced by *what* is being navigated. This raises questions about services since that is where the focus is. A valid question would be 'what aspects of a service influences navigation?' Compatibility with other services and existing platforms is one aspect visible in the scenario. The photograph editing service might for instance not be compatible with the displaying service offered by the television since the colours of the television might not support enough colours. It could also be that a service doesn't have a user interface for a certain platform, the photograph editing service might for instance not have an interface for the small display on the camera. There can be many reasons why there is no interface for a certain platform; maybe the service provider has not yet created such an interface or maybe it is simply impossible to create an interface for the service for a specific platform. An example of the latter is an interface for the photograph editing service meant for the text-based interface of a mobile phone.

Another aspect of a service influencing navigation could be the description, or metadata, of a service. The metadata of a service might help, or hinder, the user finding the right service. Alana had to choose between a few displaying services in the scenario, which she able to do based on their location. This location information was provided as metadata, but the metadata could contain different information for situations where for instance location information is unavailable or unusable. The question is whether metadata is useful for services and if so, which form it should take.

Not just the services but also the way they are *organised* has its effects on navigation. In the scenario services were organised in virtual ‘places’. These places could reflect real places, like a home or a city, but they might as well be purely virtual places, like a place containing all services relevant for a certain activity. An example of a pure virtual place was the one containing Alana’s favourite services. There are many ways to organise the services and it depends on user and the task which way will be the most useful. Geography-based organisation is just one method, another could be time-based or activity-based organisation. An appropriate organisation will help the user find the right service.

The way services are organised in turn determines the *metaphors* that can be used. For instance a location-based organisation that reflects real places allows for a geographical metaphor, which is defined as:

> ***Geographical metaphor:***
>
> *The physical characteristics of an area used to designate the characteristics of a virtual environment.*

Dieberger, (to whom we shall later return) has used a geographical metaphor (Dieberger, 1994), a city metaphor, for a large-scale virtual world. Which metaphor is being used affects the navigational task; a fitting metaphor can ease navigation.

The scenario showed a couple of situations where using a *service browser* would be a way to find the services. This raises questions regarding the browser itself. The service browser differs from a Web browser in that it can be used to *navigate services*. This means that while a Web browser mainly displays documents that in turn link to other documents, a service browser is focused on navigating *between* services. Its main task is not so much display as navigation.

The main question is what the browser should be able to do. First and foremost it should enable users to navigate easily the entire collection of services available. It should make the users task as simple as possible and prevent any errors by the user. Furthermore the service browser should be able to launch services, and render the correct user interface for the device from which the service is requested. Which interface is appropriate also depends on the situation the user is in. Furthermore service browser should also be platform independent so that users can have the same experience across different platforms. Last but not least the service browser should be multi-modal, meaning it should have multiple modes of interaction so that users can interact with it in different modes. Examples are speech mode, 2D mode or 3D mode. It depends on the situation which mode will be appropriate.

What would, or even better could, such a browser look like? The browsing experience very much depends on how the services are organised and on whether a metaphor is to be used. The scenario pointed out two solutions, a two-dimensional floor plan and a three dimensional virtual world, but more are possible. A pure text or speech based interface are two examples. The possibilities are constrained only by the way in which the services are organised. A geographical floor plan as in the scenario would be impossible if the services were not organised in places reflecting the real world. The interdependence between the organisation of services and navigational possibilities is worth a closer look.

The scenario describes a situation where the services are statically organised in places, the organisation of the services is a given and the user navigates them. Depending on location the user can access both local and non-local services. Alana for instance was able to access the services in her grandmother's house as she was physically there, but she also had access to services outside the house on the larger Internet. A different approach could be that the services are organised around the user, providing her with permanent access to a specific collection of services. She could for instance carry the collection of services around in a virtual briefcase, which is the approach adopted by the designers of the sView platform (Bylund, 2001). Whether the organisation is environment centric or user centric will have its effects on how the services are navigated.

An environment of services can be highly *dynamic* compared to the rather static environment of documents. An environment as described in the scenario consists of many services provided by devices as well as pure software services. These services can come and go depending upon whether the service provider keeps its service available. When Alana connected her camera to the network its service automatically appeared in the environment she was browsing. When she would have disconnected it again it would have automatically disappeared. This is because the camera is a service provider that upon (dis-) connection (dis-) appears. The same goes for software services, if the provider makes the service (un) available the environment of users browsing for that service changes automatically. The dynamics of the service environment puts a higher strain on the user when it comes to navigation since a constantly changing environment can be harder to understand (Theng, Thimbleby, and Jones, 1996).

## 2.4. Topic selection

Looking back at this chapter it appears that even navigating services raises too many issues to answer within the scope of this research. Thus a selection has to be made. For those specific issues a deeper investigation into relevant theories will be made. If this results in further questions an experiment can be constructed that tries to answer them.

The first topic concerns the general differences between navigating services and navigating documents. As the analyses showed it is likely that navigating services is fundamentally different from navigating documents. The preliminary literature study showed that not much research has been conducted in this area, making it a suitable topic for research.

The second topic will be the organisation of services and the possibilities for using metaphors. This is closely related to the first topic and the mentioned literature study showed that while there has been much research in spatial navigation of Information Spaces (Benyon, 1995) and the use of metaphors for such spaces (Dieberger, 1994), this is less the case of research into navigation of virtual spaces consisting of services. The organisation of this space of services might afford new ways of navigation and new uses of metaphors. Special attention will be given to geographical metaphors.

The third and last topic will be about metadata for services and its connection with navigation. There has been quite a lot of research on metadata for finding Internet resources (Kucuk, Olgun, and Sever, 2000) and an interesting question is whether there are any new requirements when discussing metadata for services.

To see whether good and relevant hypotheses about these topics can be posed we will have to look at the existing theory and research. That background is the topic of the next two chapters.

## 2.5. Summary

To narrow down the general research topic to subtopics that can be used for generating hypotheses two scenarios were worked out and analysed to see what the effects of the three changes mentioned in the introduction chapter could be. The first scenario focused on the global effects while the second scenario was developed and analysed with a special focus on navigational issues. Both scenarios generated a large amount of issues that lend themselves for further research, but only three topics were selected for this thesis. The first topic concerns the general difference between navigating services and navigating documents. The second topic is about the organisation and metaphors that can be used with services, with a special focus on geographical metaphors. The third topic is about the metadata requirements for services. To find out what has and hasn't yet been researched within the confines of these topics the following chapter takes a look at the existing body of theory about navigation. Subsequently the findings of this chapter will be reflected in light of those theories to see where they fit and whether there are any open issues.

# 3. Navigation

*Navigation is the focus of this research. This chapter provides the theoretical framework from which, in conjunction with the findings of the previous chapter, relevant hypotheses can be constructed. It discusses theories of navigation in general and navigation of Information Spaces in particular. To structure the discussion a hierarchy of navigational aspects will be introduced.*

In the previous chapters we saw the effect of the move towards a network of distributed services and documents accessible from multiple devices. Tomorrow's computer systems will no longer consist primarily of stand-alone applications processing locally stored documents developed for known users in a known setting. Rather the computer systems will exist of distributed documents and services, available to a distributed user group. Similarly users will no longer be faced with one or more dedicated local applications, but rather with a dynamic and extensive Information Space consisting both of documents and services. The services will be platform independent and commute between platforms. Being no longer bound to the desktop computer, they will run on diverse hardware platforms – ranging from small devices to powerful network servers. This trend has already started. These developments raise the question of whether finding the services and documents using different devices introduce new navigational pitfalls and/or opportunities.

Before these issues can be addressed it is necessary to examine existing theories of navigation and see whether they are applicable to the new situation. The four research areas mentioned in the introductory chapter[41] each have valuable theories on the subject of navigation and will be addressed in this chapter. First we will deepen our exploration of the general concepts of navigation. This will be followed by a discussion of what constitutes an Information Space. The subsequent section will introduce the hierarchy of navigation that gives an overview of the relevant aspects of Information Space navigation.

---

[41] Human-Computer Interaction, Distributed Computing, Knowledge Technologies, and Ubiquitous/Pervasive Computing.

## 3.1. Navigation

In his seminal 'The image of the city' (Lynch, 1960), which served as the starting point for a long research tradition, Kevin Lynch investigated navigation of cities as physical environments. Lynch's focus was on developing guidelines for constructing easy navigable cities. He did this by studying how people build up a mental image of the city they are navigating and identified five distinct elements within this mental image;

- Paths.
- Edges.
- Nodes.
- Districts.
- Landmarks.

As these terms have been seminal to research in navigation and later of the navigation of Information Spaces a brief summary of each term's meaning is given in the paragraphs below.

Lynch used the term *path* for the connections along which the user moves between city elements. They result from the limitations imposed upon movement by the city's structure. Other than within open space, movement in cities is restricted by the way in which the city is built. Looking at a city map one can easily see that the city's structure allows only certain paths. As a path is an element of users' mental image of the city it can consist not only of visual sensory inputs, but also of the streets' smell, the feelings the user experiences upon seeing buildings she likes, or the street's texture.

Just as paths connect them, *edges* separate individual city elements. Examples are neighbourhoods that visually differ from each other, or a street that a user wants to cross. At times, as with crossing a street, these edges can be crossed at others not, as with a wall between two houses.

*Nodes* occur at the intersections of linear elements such as paths and edges. Whereas paths and edges are linear elements nodes are points that can be entered. Examples of nodes are crossings, public squares and roundabouts.

*Districts* are areas combining the other elements, including itself. Well-known city districts are for instance Manhattan and Brooklyn in New York. But districts might also be defined by the function of an area, such as a university campus or a harbour. What is different about districts is that may not be mainly visual. Smell, as with a harbour, might be equally or more important.

The final element of the mental picture is the *landmark*. A landmark is a unique characteristic of an environment, and can differ widely in both size and form, for example a statue, or a skyline. They are easily recognisable and distinct from the environment and can help the user find her way.

Taken together the five elements help humans navigate a physical environment. Lynch's description of the elements can be found in later descriptions of navigation, which largely build on the same concepts (see for instance (Darken, 1996)). Trailblazing, for example, describes creating a path through unknown territory. In practice this mainly meant accurately marking land resulting in a trail of landmarks, a concept identified by Lynch pointing up the importance of his work for the field of navigation.

The five elements have also proven to be useful for the initial research into navigating non-physical environments (Darken, 1996;Dieberger, 1994). Lynch's findings were interpreted in the context of navigating Information Spaces, resulting in some useful definitions of navigation of this new type of space. The most useful definition in the context of this thesis, and therefore adopted here, is given by Benyon & Höök in 'Navigation in Information Space' (Benyon and Höök, 1997). They define navigation as consisting of three types of activity; wayfinding, exploring and object identification.

Benyon & Höök first identify the more traditional *wayfinding activities* as part of the definition of navigation. Wayfinding is defined by Peponis et al as "the ability to find a way to a particular location in an expedient manner and to recognise the destination when reached" (Peponis, Zimring, and Choi, 1990, 555-590). According to Downs and Stea (Downs and Stea, 1973) this process can be described as a four-step process:

- Orienting oneself in the environment.
- Choosing the correct route.
- Monitoring this route.
- Recognising that the destination has been reached.[42]

Wayfinding stresses the cognitive aspects of planning a route. A point recognised by Darken (Darken, 1996), who basing himself upon Passini's conception of wayfinding as a facilitation of navigation (Passini, 1984), acknowledges this by using the term 'navigation' for directed movement action and 'wayfinding' for cognitive action involving route determination. Both cognitive and physical aspects are relevant for the

[42] With wayfinding the assumption is that the actor knows where they want to go, which sets it apart from the other two activities.

study of navigation within the context of this thesis, and I therefore adopt Benyon & Höök's definition of navigation that incorporates wayfinding activities.

Secondly according to Benyon & Höök navigation can be an *exploring activity*. When exploring people are not so much trying to find their way but are 'just looking around.' They are trying to grasp their environment. The difference between this and wayfinding is that the person is not yet sure where to go.

The third activity of navigation is *object identification* according to Benyon & Höök. In this case the user is not so much interested in the location of an object or in finding a path or reaching a goal, but more in finding categories and clusters of objects, finding interesting configurations of objects and finding out information about the objects.

Together the three activities cover all of the navigational tasks relevant to this thesis. Benyon & Höök's definition of navigation covers the lexical definition, stressing both the planning aspects, as well as the more tentative aspects of wayfinding. It also covers both physical movements and cognitive action involved in navigation.

## 3.2. Information space

The other key concept is 'Information Space'. Unfortunately there is no single standard definition of what an Information Space is, even Munro & Höök's 'Social navigating of Information Space' (Munro, Höök, and Benyon, 1999) lacks it. The literature study showed that most of the research assumes a common understanding of what an Information Space is. Information Spaces have been researched extensively within the field of *Computer Supported Cooperative Work* (CSCW) - one of the first research areas to use the concept. The original description used was as broad as "a 'space' comprising data, personal beliefs, shared concepts, professional heuristics etc." (Bannon and Schmidt, 1989). It was broadened even further with the observation that an Information Space not only consisted of the information objects but also of the joint interpretation of these objects by the actors involved (Schmidt and Bannon, 1992, 7-40).

As Redström et al (Redström and others, 1999) point out problems arise from the broad connotations of both 'information' and 'space'[43]. However they give a working definition:

> *"The information available at a given location, as perceived by a user or device".*

[43] For an excellent account of the meaning of the concept 'space' throughout history I recommend Margaret Wertheim's 'Pearly Gates of Cyberspace' (Wertheim, 1999). It certainly is relevant in light of this thesis since it also treats virtual spaces like cyberspace.

They point out that this definition applies to information in both the real and virtual worlds as physical environments such as computer networks can mediate Information Spaces. They settle for a description that separates *digital* from *non-digital* Information Spaces. That separation is important for this thesis as well since the focus is on services made available through computer networks. Therefore I will make the separation explicit in the working definition used here. Furthermore the focus is on humans navigating Information Space, and not on devices, which will therefore be left out of the working definition.

> ***Information space****:*
> *The information available at a given location through digital media, as perceived by a user.*

Where 'information' is defined as:

> ***Information****:*
> *Processed, stored, or transmitted data.*

Examples of Information Spaces could be the virtual world perceived by a user playing a three-dimensional game on a computer, and a physical surrounding augmented with devices providing digital information[44].

Benyon & Höök (Benyon and Höök, 1997) tackled the problem of the distinction between real and virtual places by an interpretation of the concept of Information Space that includes real world spaces as well as augmented and virtual worlds. Real world spaces are included since they are overlaid with artificial information such as street signs, advertisements or traffic announcements. However real world places without digital information are excluded by the working definition given above and as such form no part of this research project.

*Augmented worlds* and *virtual worlds* are both included by the working definition of an Information Space. An *augmented* world is a physical place extended with a layer of information and functionality made accessible *via* digital media. Examples of such augmentation might be screens on refrigerators giving recipe information from the Internet, or a button on a television remote control that lets you check your e-mail. *Virtual* worlds on the other hand are spaces that have no ties to physical places. Of which perhaps the best-known examples are online multiplayer games and the Web. Both types of Information Spaces are subject of this thesis.

The separation between augmented and virtual worlds is not very strict; rather it resembles a sliding scale. It is hard to pinpoint where the augmented world turns into a

---

[44] Where 'digital information' means information offered by digital media.

virtual world and vice versa. Some augmented worlds have few virtual components[45] while others contain almost nothing else (Azuma, 1995, 355-385). How many virtual objects does a physical world need before it turns into a virtual world? Besides that a virtual world can have connections to the physical environment the user is in while navigating the virtual world (Streitz and others, 1998). An example would be changes in the virtual environment depending upon whether the user is entering it while being physically located in a kitchen or in an office. So augmented and virtual worlds differ in the number of virtual components, but what they have in common is that they give the user access to digital information and services.

In their 1999 paper 'Scalable and Flexible Location-based Services for Ubiquitous Information Access' Jose & Davies (Caswell and Debaty, 2002;Jose and Davies, 1999), give a nice example of the virtual and augmented worlds coming together in presenting a generic architecture for building location-based applications. They achieve this by combining physical location with semantic information. Within the CoolTown project[46] this idea was further elaborated by labelling places with URLs[47] (Caswell and Debaty, 2002;Pradhan, 2002). This way real world places become accessible *via* the Web, users can enter the URL of such a place in the Web browser and get the Web page from that place.

With the advance of powerful computers and large interconnected data networks such as the Internet virtual worlds came under the scrutiny of the research community (Steed, 1993). For instance technical questions as how to create virtual worlds were addressed by the Computer Science community, while the Human-Computer Interaction research community investigated the usability aspects of such worlds when used for performing certain task. Augmented worlds however required more technology to be in place than virtual worlds and thus came into focus much later. Augmented worlds had higher requirements from research fields such as ubiquitous/pervasive computing and distributed computing, (see section 1.3). What was needed for the creation of virtual worlds was no more than one powerful computer. Later these computers could be connected to create shared virtual worlds. Augmented environments on the other hand required devices that offer the users access to digital information and services - which is the topic of pervasive computing, as well as devices that can take over some tasks from the user - which is the topic of ubiquitous computing. This required much more technology to be in place. The devices, including their software, had to be developed. Besides that most of the devices had to be interconnected for exchanging information with the rest of environment (where the environment could be as large as the rest of the world). This raised many issues by itself of which many are covered by the research field of

[45] Information and functionality made accessible *via* digital media.

[46] See http://www.cooltown.hp.com

[47] Uniform Resource Locator

distributed computing, which explores questions as how these interconnected devices can (spontaneously) work together, how they can run autonomously, how functionality can be distributed across a network and so forth.

It was not until the technology and science for creating these networks of interconnected devices was in place that the virtual world started merging with the physical environment of the user, resulting in what is now known as augmented reality. Thus in retrospect we can see that knowledge about distributed computing and technological advances in the area of pervasive/ubiquitous computing opened up new possibilities for creating Information Spaces. And, as is usual with scientific advances, each new opportunity raises new questions, such as, for instance, how to navigate Information Spaces.

## 3.3. Navigating Information Space

Once the pioneers in the field of computers science discerned the coming into existence of this new and strange concept of Information Space, they raised questions as to how this space should be presented to the user. In his visionary 1945 article 'As we may think' Vannevar Bush (Bush, 1945, 202-208) raised the question of how users might navigate large amounts of data. He proposed a theoretical information processor called 'Memex' that enabled users to thread through massive amounts of data, strikingly similar to the way users surf the Web now. Since Bush was not talking about digital information[48] the large amount of data he mentions does not constitute an Information Space as defined above. Steven Johnson discussed the Memex in his 'Interface Culture[49]' (Johnson, 1997) saying it "was designed to organise information in the most intuitive way possible, based not on file cabinets or superhighways but on our usual habits of thinking-following leads, making connections, building trails of thought". This shows that Bush was confronting an organisational as well as a navigational problem.

It wasn't until advances in computer science enabled storage of large amounts of data that navigation became real problem[50]; those advances enabled the first Information Spaces. In 1968 Douglas C. Engelbart and the group of 17 researchers working with him in the Augmentation Research Centre at Stanford Research Institute,

---

[48] Digital information would become more relevant later on.

[49] A book written for the general audience about the way technology and art come together in developing the human interface to Information Space. While not a scientific work in itself it does provide a good overview of the state of the art in research.

[50] It is hard to give a qualitative measurement of when navigation becomes problematic; it depends on many factors, of which for instance the user is one. A good short-term memory and experience with navigating a certain collection of data will for example be helpful. It is a sliding scale; a small collection of data will give no troubles for navigating, while a very large collection will be problematic for most users.

demonstrated a new online system called NLS[51], which they had been working upon since 1962[52]. NLS was the first attempt at making an interface between the user and the collection of data inside the computer or on the network[53]. In 'Interface Culture' (Johnson, 1997) Steve Johnson stresses the importance of this moment and puts it into a larger perspective. He tells the story of how the first interface designers, like Engelbart and his team, had trouble devising a way of organising this new virtual 'space' consisting of data. Initially there was only one requirement, according to Johnson; "You could build anything you wanted in that new information-space, but it had to be simple". The problem was that this new space was uncharted territory. There was no clear understanding what it was, what it offered, or where it was going, making it very hard to construct a simple interface between this space and the users[54].

In the decades after Engelbart's demonstration computers became more popular and users slowly grew accustomed to the idea of navigating large amounts of digital information (Information Space)[55]. The growing interest also meant that Information Space more often became a topic of research; from the technical questions of how to construct Information Spaces to humanistic questions of how to make Information Space easy to grasp for humans (see for instance (Hamit, 1993;Mazuryk and Gervautz, 1996)). Navigation theories have especially been applied extensively to navigating Information Spaces (Dieberger, 1994), resulting in a long research tradition. The problem is that many discussions in that field mix their terminology[56] making it hard to compare their results. So to avoid confusion and make clear what I am talking about it is useful at this point to provide an overview of relevant concepts and how they are related. For that purpose I developed a hierarchy (shown below) that orders all variables involved in navigating Information Spaces. Although the hierarchy doesn't provide an exhaustive overview of examples within each level, it does mention all the levels at play when discussing navigating Information Spaces.

---

[51] On-line System.

[52] A video of the complete demonstration can be viewed online here:
http://sloan.stanford.edu/MouseSite/1968Demo.html

[53] From here on I will use the definition of 'Information Space' as given in the previous section.

[54] Johnson continues with a rather interesting discussion about the influence the design of this interface has on our daily lives, but that is unfortunately outside the scope of this thesis.

[55] They weren't aware of the concept of Information Space, which was invented in hindsight.

[56] See for instance (Munro, Höök, and Benyon, 1999), e.g. paragraph 2.3, for a discussion of the confusion surrounding this terminology. Another source is (Dourish and Chalmers, 1994).

| | |
|---|---|
| **Navigation technique** | … \| Spatial \| Social \| Semantic \| … |
| **Mode** | … \| Text \| 2D \| 3D \| Speech \| Haptic \| … |
| **Metaphor** | … \| Desktop \| Geography \| Pile \| Window \| … |
| **Organisation** | … \| Time \| Activity \| Semantics \| Location \| … |
| **Entity** | … \| Documents \| Services \| … |

**Figure 3-1: Hierarchy of navigation**

There are five levels involved in navigating Information Spaces. Only the order of the levels is fixed, but there are multiple paths from bottom to top. For instance there can be documents organised by time using a calendar metaphor, which is rendered to the user as a 2D interface. Another example could be applications that are geographically organised using a city metaphor and which is rendered to the user as a 3D virtual reality system. The user could use spatial navigation to move through Information Space. However not all combinations are useful (or even possible). The following chapter will discuss existing navigational systems and their combinations.

The small vertical lines between the different concepts within each layer indicate that the horizontal separation is not as strict as the vertical separation. Concepts often overlap or can be combined. An example is a virtual reality world where the user can read a document by opening a regular window, which is a combination of a 2D and a 3D interface.

The dots at each level indicate that there are probably more categories than specified in this overview. There are more metaphors one can think of for instance, but for clarity's sake a couple will do. So the hierarchy of navigation is complete in vertical manner, but not in a horizontal manner.

Together the five layers define how an Information Space is built up and the ways it can be navigated. What follows is a deeper analysis of each of the levels in a separate section, starting at the bottom.

## 3.4. Entity

The first layer seen from the bottom is concerned with whether it is information (a document,) or functionality (a service,) that is being offered[57]. From a user's perspective the question would be "what am I navigating?" Although services are not part of the working definition of an Information Space, they are mentioned here to stress the fact that the entities navigated can be more than merely documents. The next chapter will explore services and Information Spaces in more detail; this chapter will focus on navigating documents.

Theories about navigating hypertext systems (or hypermedia) are based upon the fact that what is being navigated is documents, see for instance (Conklin, 1987;McKnight, Dillon, and Richardson, 1991;Neumueller, 2001;Nielsen, 1991). Hypertext was first defined by Ted Nelson in 1967 (Nelson, 1967) as "a combination of natural language text with the computer's capacity for interactive branching, or dynamic display ... of a non-linear text ... which cannot be printed conveniently on a conventional page". Most of the early hypertext systems were incompatible so a reference hypertext model, the Dexter Hypertext Reference Model, was defined (Halasz and Schwartz, 1990). The Dexter model tried to "capture, both formally and informally, the important abstractions found in a wide range of existing hypertext systems and future hypertext systems". The model proved useful[58], but much of the research in navigating hypertext systems is from the second half of the nineteen nineties, when the most successful application of hypertext, the World Wide Web, was first being introduced (Berners-Lee, 1989;Conklin, 1987;Berners-Lee and Fischetti, 1999).

It is understandable that the first research on navigating Information Spaces dealt with documents. The technology for offering more than just documents *via* the Internet, as discussed in section 3.2, was only starting to become available[59]. The consequence was that all the layers in the hierarchy of navigation on top of the entity layer only *assumed* a collection of documents. This affected the metaphors that were deemed fitting, the well-known desktop metaphor is a good example of this, and shows that the focus was on documents rather than functionality.

---

[57] Nielsen calls them 'information objects' and 'functionality objects' in his newsletter (Nielsen, 2000a). I will continue to use 'information' and 'service'.

[58] See for instance work by Kaj Grønbæk (interesting follow-up research is presented in (Grønbæk and Trigg, 1996, 149-160)).

[59] There actually are a few examples of services that *were* available at that time, but they were rather exotic and not mainstream, for instance remotely controlling computers through the Internet by using Telnet. The initial Telnet protocol specification is from 1972 (Postel, 1972).

The focus of this thesis is mainly concerned with this difference in entities and the way in which it affects the rest of the hierarchy of navigation.

## 3.5. Organisation

The next level is concerned with how the documents are organised. It depends on the situation who controls the organisation, either the provider or the consumer of the documents.

In the case of a standalone computer the owners are the providers of the documents and decide how they organise their documents[60]. The usual way is to create a tree-like folder structure on the hard disk of the computer and store documents within it. It doesn't matter whether the tree structure is viewed using a command line interface or a graphical interface; the organisation of the documents stays the same. The situation is largely the same for the documents of a Web site accept that access to a Web server is needed for this and most users, besides Web developers, don't have that[61]. The ones that do have access to a Web server can decide how the documents on their Website are organised. They are the providers of the documents and their organisation of the Web pages is what the consumers are confronted with.

Even though the providers might have organised their documents in a certain way, the consumer of the documents doesn't have to use that organisation. A Web site can for instance be a coherently organised collection of Web pages, but if a user decides to access the separate pages by using bookmarks the organisation is replaced by a user defined organisation. The same goes for documents on a standalone desktop computer. Even though they might be organised in folders by topic, the consumer might choose to view the documents using a different view, for example a using a search application. Sometimes a specific application helps the user organise the documents. In case of the Web this can be a directory (such as Yahoo![62]) or a Web browser with the ability to create bookmarks.

It should be said that in case of the Web the providers have only limited control over the overall organisation since each of them only controls the organisation of a very small subset of all the documents. The fact that there is no one overall organising

---

[60] Exceptions are when applications store files automatically.

[61] With the current hype surrounding Weblogs this is slowly changing. A Weblog, sometimes called 'blog', is an attempt to implement Tim Berners-Lee's initial vision of a Web that is a writing environment as well (Berners-Lee, 1989). It is a personal chronological log on the Web that is updated on a frequent basis and can cover an unlimited range of subjects. It often contains links to other Web pages. Sometimes it even contains pieces of text from other Web pages supplemented with comments. See http://www.weblogs.com for a small listing of recently updated Weblogs.

[62] http://www.yahoo.com

principle is a source of many of the problems with Web navigation (Theng, Thimbleby, and Jones, 1996). Attempts to solve this problem include Web directories and search engines (like Google[63]). Web directories try to superimpose a structure onto the existing Web instead of requiring providers to organise documents using their own hierarchy. The result is that existing directories are by definition incomplete and out of date. Search engines don't try to organise the Web but enable searching the almost complete collection of documents at once. It allows the user to view a collection of documents containing the same keywords. The problem with this approach is that a few keywords often yield so many matches[64] that even though the total number of possible documents has significantly shrunk the result set still requires some effort from users to find exactly what they are looking for.

What these examples show is the variation in how documents are organised and in who controls the way in which they are organised. A similar situation occurs for applications. If we look at applications on stand-alone desktop computers we see that the user decides how to organise them[65]. The most common method in current operating systems is a hierarchical menu containing applications (like the highly similar 'start menus' under Windows, Linux and Apple Macintosh) combined with the possibility of storing them on the desktop. The desktop in such a case is the initial view offered when starting up the operating system[66]. The start menus and desktop both allow for modification by the user.

With the advent of services offered *via* the Internet the power to decide on the organisation could once again be in the hands of the provider, who might, for example, decide to offer a collection of services organised around a theme such as office productivity. It should be said however that system flexibility is a key influence here, in a flexible system users might have the freedom to organise the service according to their preferences.

Next we will look deeper into a few well-researched ways of organising documents, there are likely more but these will cover most of the existing ways.

---

[63] http://www.google.com

[64] Based on personal experience. The problem of describing a document in keywords is closely related to the problem of extracting the meaning from a document purely based on its content as discussed in section 2.2.1.

[65] Most of the existing applications propose a default location on installation, but in all cases this location can be changed by the user.

[66] The desktop has no resemblance to a "real" desktop anymore, but it did with the first versions of the graphical interfaces for operating systems.

### 3.5.1. Time-based organisation

One way to organise documents and services is chronological. This can for instance be the last time a document was edited or a service was used. There has been some research on time based organisation of documents, and the LifeStreams project is a good example (Gelernter, Fertig, and Freeman, 1996;Freeman and Fertig, 1995;Freeman, ). LifeStreams orders documents in so-called time-ordered streams and is meant to replace conventional files and directories (Freeman and Fertig, 1995). The graphic below is a screenshot of LifeStreams, and shows a stream of documents together with one separate document upon which the user is working. The scrollbar at the bottom of the screen tells what time span the stream covers. Lifestreams uses colours and animations to highlight important features of documents. It also offers the opportunity to create and view sub streams that cover specific subjects.

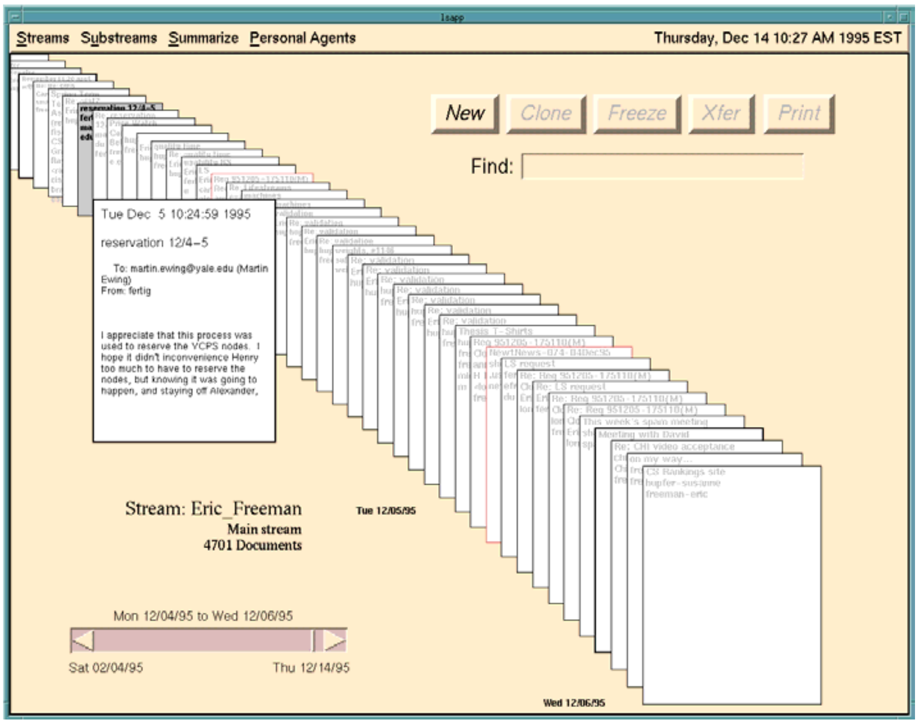



**Figure 3-2: A screenshot of Lifestreams**

Freeman and Fertig give three reasons for using a time-based organisation. First of all they say it adds a historical context to the documents, meaning that documents have an association with other documents created, edited or accessed around the same time. This can for instance be crucial in a organisational setting (Cook, 1994). Secondly the present portion of the stream can act as a workspace providing the user access to all of the documents upon which she is working. Thirdly the future portion of the stream can act as a scheduler by allowing the user to post documents in the future.

Rekimoto (Rekimoto, 1999, 45-54) proposed another concept for the time-centric approach to organising information on computers called ‘time-machine computing’. It is based on the same time-ordering concept as the LifeStreams system but combines it with a spatial arrangement. The disadvantage with pure time-based systems he says, is that they neglect the activity status (what the user was doing) at the time the document was created, making them more suitable for archiving data than for providing workspaces (Rekimoto, 1999, 45-54). Rekimoto assumes that the activity status is reflected by the spatial organisation of the documents on the desktop of the computer. So by storing the history of the desktop and making that available to the user he combines chronological navigation with a history of the activity status, resulting in a prototype that is like a time machine for the information environment. An implementation of the concept is called TimeScape.


**Figure 3-3: Screenshot of the TimeScape desktop**

TimeScape has no folders and shows all items (documents as well as applications) on the desktop. On the left hand side you can see a timescale and using the dial on the top or the back and forth buttons users can navigate in time. The desktop will revert to the state it was in at that specific time.

Yet another project experimenting with time-based organisation of information is the LifeLines project at the university of Maryland (Plaisant and others, 1998, 76-80).

LifeLines is a ‘general visualisation environment for personal histories’. LifeLines offers many sophisticated ways of navigating the medical records of patients, but time was chosen as the overall way the records are organised. Medical records are very information heavy and the problem is providing doctors with easy access to the whole collection of information. Here is a screenshot that shows medical records:

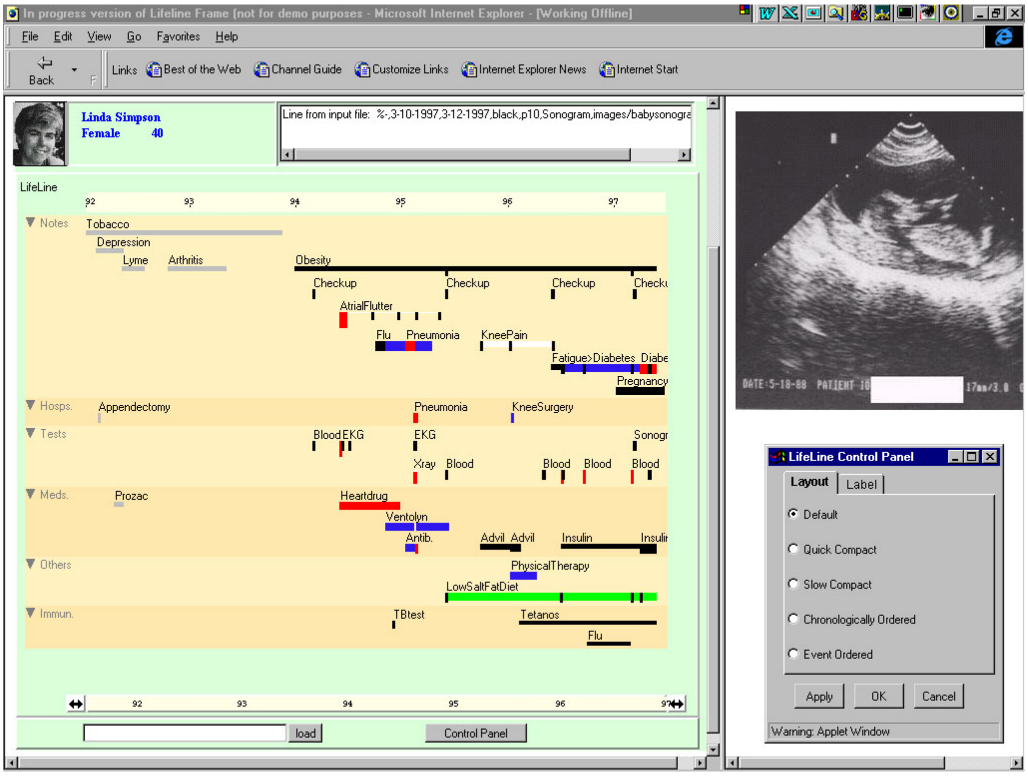



**Figure 3-4: Screenshot of LifeLines**

The main difference with LifeLines compared to the TimeScape system is that the items placed along the timeline are not separate documents. They are descriptions that in turn are organised along the vertical axis. So regarding the *spatial* organisation one could say the LifeLines lies somewhere between LifeStreams and TimeScape.

As these examples show a time-based organisation of information is a viable way for providing users with a way to navigate large amounts of data.

### 3.5.2. Activity-based organisation

Yet another way to organise information is by activity. An example would be documents grouped by whether they are related to ones work or to ones private life. There are several approaches to this way of organising information.

Task-based partitioning of user workspace was the first research topic addressing information organisation based upon activities. The classic reference is Henderson and Card's work on 'Rooms' (Henderson and Card, 1986, 211-243) where they describe a way to reduce the overhead in task switching by mapping each activity to a virtual desktop. Each desktop in turn contains a set of windows[67] that are relevant for the activity. Later the CSCW[68] research community adopted this partitioning of workspaces (see for example (Robertson and Roseman, 1998)) and applied the idea to *shared* workspaces. Recently a 3D visualisation of 'rooms' was developed by Robertson et al (Robertson and others, 2000).

Schneiderman & Plaisant took another approach when introducing the *personal role manager* (Plaisant and Schneiderman, 1995;Schneiderman and Plaisant, 1994). The key to personal *role* managers is that information is organised according to the *role* a person plays in an organisation. When working in a *role* the person has most relevant information visually available. These visual cues remind her of her goals, the related individuals, required tasks and scheduled events.

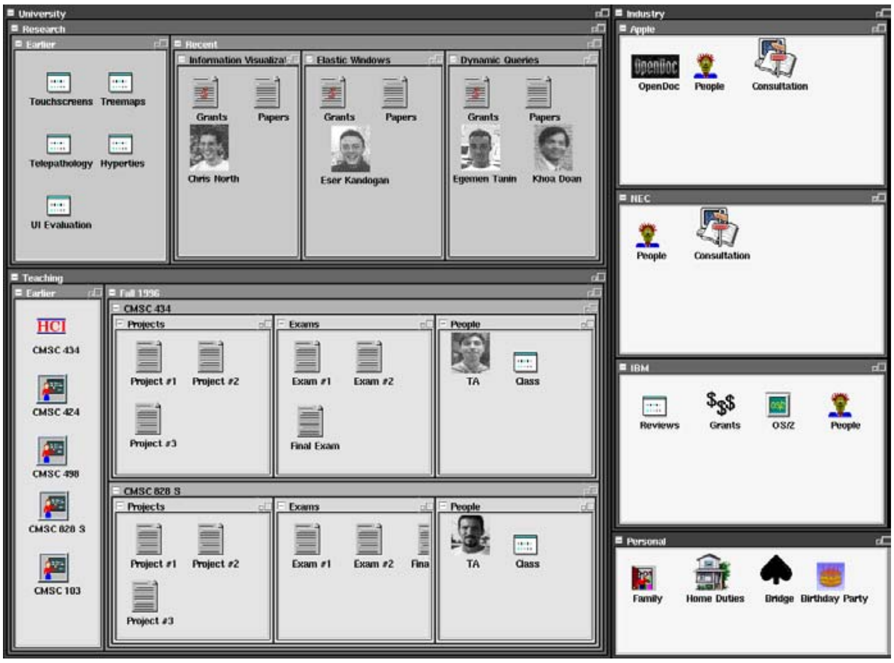



**Figure 3-5: A Personal *Role* Manager (Kandogan and Schneiderman, 1997).**

---

[67] A scrollable viewing area on screen.

[68] Computer Supported Cooperative Work

Dourish et al (Dourish and others, 1999a) investigate how the *workflow*[69] may be used to organise information. Some users use a very strict workflow and organising their information in the same way is an option. How useful this approach is depends on the user. Users with a strict workflow will find it more useful than users with a less strict workflow.

One last approach to activity-based organisation is Presto, a "prototype document management system providing rich interaction with documents through meaningful, user-level document attributes, such as 'Word file', 'published paper', 'shared with Jim', 'about Presto' or 'currently in progress'" (Dourish and others, 1999b). An interesting aspect is that metadata is an important aspect of Presto since document attributes describe the different roles a document might play.

### 3.5.3. Semantic organisation

Semantics is the study of the relationship between words and meanings. The field of semantics has three basic concerns: the relations of words to the objects denoted by them, the relations of words to the interpreters of them, and, in symbolic logic, the formal relations of signs to one another (syntax) (Columbia University, 1999). When such relationships are used to organising information it is called a semantic organisation. A simple but good example might be the organisation of documents on a computer in folders representing different topics. The folders can in turn contain subfolders together building a semantic hierarchy.

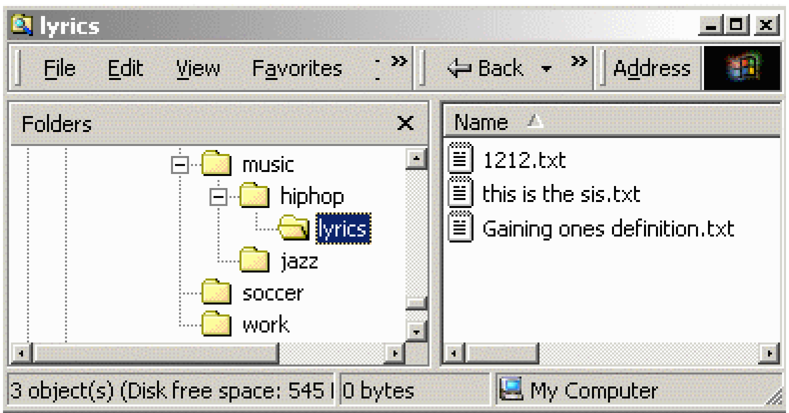



**Figure 3-6: A semantic organisation of documents.**

[69] Which they call "a formal representation of well-defined activity" (Dourish and others, 1999a).

Another example could be a Web page containing links to other Web pages on a certain topic. Such a page in that case organises these links based upon their semantics.

A semantic organisation is often used without it being very obvious. Steven Johnson questions (Johnson, 1997) whether files and folders have real fixed spatial location in the world, although they look like they do on the desktops of today's most common operating systems[70]. He claims that "…spatial coordinates on the desktop are simply an illusion…" The organisation of files and folder is based on semantics instead of spatial coordinates. In particular the fact that the files and folders are often found by using other ways of navigating than browsing through a folder structure illustrates Johnson's point. Files can be searched using a search tool, they can be accessible from a list of recently accessed files or they can be accessible in a application specific way. This shows that files have different meanings in different contexts, constantly changing the way they are semantically organised, irrespective of whether users look at it using a text interface or a folder hierarchy. In such a scenario a fixed physical location as suggested by the desktop of current operating systems is an illusion.

Semantics is a topic close to the core of projects such as the Semantic Web (McIlraith and others, 2001, 46-53;Berners-Lee, Hendler, and Lassila, 1 A.D.) and others within the field of knowledge technologies[71]. According to Tim Berners-Lee "the Semantic Web is an extension of the current Web in which information is given well-defined meaning...", which stresses their focus on a semantic way of organising information. Automating this process would require something like a semantic layer on top of Information Space, a layer of metadata describing the information.

### 3.5.4. Location based organisation

Another way to organise information is by similarities in location. An example of this is a collection of all the information generated in a factory. A location based *organisation* only states that the information is organised by the real location of the information, not how it is presented to the user. It doesn't say whether the user interface is a 3D virtual reality world or a text interface. It's up to the '*mode*' layer in the hierarchy of navigation[72] to specify that. Information can for instance be organised by location without using a 3D interface mode to navigate it.

To conclude this sample of different ways of organising information I must again stress that these are not the only possible solutions, it is merely a to highlight the most important aspects. Another important point to stress is that these different techniques

---

[70] Linux, MacOS, Linux.

[71] See the introduction chapter for an overview of those projects.

[72] See section 3.3.

are often combined. Thus in the LifeLines system a time-based organisation was combined with organisation by patient. While important the distinction mainly serves to facilitate discussions about different ways of organising information.

## 3.6. Metaphor

A metaphor can be used to increase the user understanding of the organisation of information. This is the next layer in the navigation hierarchy and the only one that is optional since it is not always useful or even possible to use a metaphor. Well-known examples are the desktop metaphor for the operating system and the notebook metaphor for a text editor.

The dictionary definition of a metaphor is:

> *A figure of speech in which a word or phrase that ordinarily designates one thing is used to designate another, thus making an implicit comparison*
> *- (American Heritage, 1996)*

Lakoff & Johnson (Lakoff and Johnson, 1980) extended this definition by arguing that a metaphor is not just a figure of speech but also an essentially cognitive process. It incorporates bodily and socio-cultural experience in defining concepts. This means that humans use metaphors to link two domains together but this relationship might be influenced by knowledge from other domains.

Although the idea behind a metaphor is to enhance understanding, discussions in the Human-Computer Interaction research community show that this is still a topic of much debate. Donald Norman for example thinks that “a metaphor is always wrong, by definition” (Norman, 1998). Furthermore in ‘Social Navigation of Information Space’ (Munro, Höök, and Benyon, 1999) Andreas Dieberger highlights the problem that it is still unknown how well social connotations map from the physical to the virtual space. This incomplete understanding of the mapping increases the chance that a metaphor doesn’t completely convey the social aspects of navigating Information Spaces. When this occurs the metaphor might either raise false expectations, obscure real opportunities, or both. The problem is that no metaphor exactly conveys the properties of a system; if it did it wouldn’t be a metaphor anymore but the system itself. A good example of the unfortunate use of a metaphor is Microsoft Bob (Fries, 1995). Introduced in 1995 as a user-friendly alternative to Windows Microsoft Bob promised a living room on every desktop, filled with a group of animate agents that would help the user get around their computer. What was problematic was that these agents looked too much like real living characters, raising expectations that a computer agent could never fulfil. And instead of simulating a living room Microsoft Bob tried to be one, raising the expectations that you were in a real living room with real people and should act accordingly (Fries, 1995). One of the developers recognised this problem in hindsight saying that the agent "needed [to] be an interface where the character was more a part of the normal Windows user interface"[73].

Metaphors *are* being used however and they deserve a place in the hierarchy of navigation. It's up to the designers and users whether they think a metaphor is useful or not.

Metaphors can prove to be useful when explaining a new and unknown concept, this is why they were used to make users familiar with the newfound territory of Information Space. Steve Johnson (Johnson, 1997) clearly explains the opportunities and dilemmas of the early days:

> *The bitmapping revolution[74] had introduced the concept of dataspace[75], but it was still mostly a tabula rasa, an empty lot waiting to be filled. What were the new information architects going to build on that real estate? A strange mix of open-endedness and limitation was contained in the question. Because the computer was by definition so malleable, capable of shape-shifting from one visual metaphor to another, it was theoretically possible for the interface to look like practically anything: a house, a factory, a movie, a diary. But the limitations of seventies technology […] meant that flights of fancy would quickly bump up against the ceiling of hardware shortcomings.*

The metaphor chosen was that of the desktop and the effects of that choice can still be seen in today's computer systems.

Metaphors can be used at different levels; from describing a very large Information Space using a city metaphor[76] to describing just one application using a typewriter metaphor.

A metaphor is optional and some systems such as experimental information visualisation systems use no metaphor at all[77]. However when it comes to navigating

---

[73] Tandy Trower in an online interview: http://www.agentry.net/AgentryInterview.asp

[74] The invention of a graphical interface to give users access to data and applications on the computer.

[75] Johnson sometimes uses this term for Information Space.

[76] See Dieberger (Dieberger, 1994).

[77] 'Readings in Information Visualization' (Card, Mackinlay, and Shneiderman, 1999) has many good examples. For a large collection of online examples see the CyberAtlas:

http://www.cybergeography.org/atlas/atlas.html

Information Space(s) metaphors seem to have been rather popular (Card, Mackinlay, and Shneiderman, 1999). In the paragraphs below I will discuss a few examples to highlight the points I consider important.

## Geographical metaphor

A geographical metaphor uses the real physical world (geographic space) to help users grasp new territory. It is mainly used for large-scale spaces (Kuhn and Blumenthal, 1996). The distinction between large-scale and small-scale spaces comes from the comparison to the human body. Small-scale spaces contain objects that are comparable in size to the human body and can easily be moved and turned (Kuhn and Blumenthal, 1996). Objects in large-scale space are much bigger than the human body and have fixed positions, such as buildings, streets etcetera.

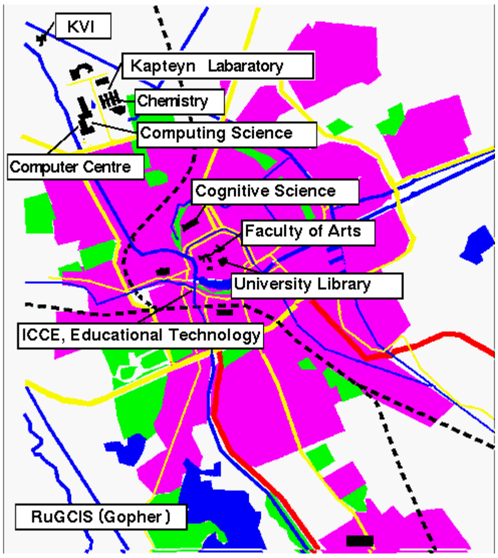



**Figure 3-7: Map of a town to organise information: a geographical metaphor for Information Space.**

Geographical metaphors are often used to help users navigate Information Space. Trying to map the human ability to navigate geographical space onto Information Space has had varying success though (Munro, Höök, and Benyon, 1999). As we have seen a fundamental problem of metaphors is that there is often a mismatch between what the possibilities are and what the metaphor communicates.

When using a geographical metaphor a designer should be aware that there is a difference between physical location and geographical location (Caswell and Debaty, 2002). They both point at a place in this world, but the former specifies the position of it by a global coordination system while the latter organises places in a hierarchically manner. GPS[78] coordinates are an example of specifying a physical location while Europe/Netherlands/Groningen would specify a geographical location. Both types of location can be used for metaphors.

## Desktop metaphor

A much-discussed metaphor is the desktop metaphor. It was the Xerox 8010 "Star" Information System that had the first graphical user interface with a desktop metaphor. It had icons, dialog boxes, universal commands, and a "point-and-click" style of interaction now known as "direct manipulation".

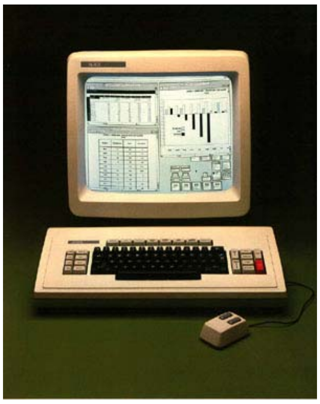


**Figure 3-8: Xerox Star 8010, the first to use a desktop metaphor**

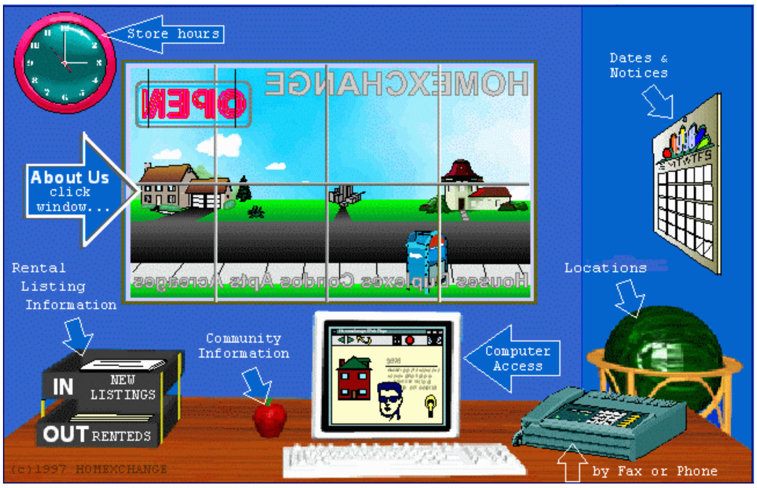



**Figure 3-9: The initial idea of a desktop metaphor.**

---

[78] Global Positioning System.

Initially the desktop was a fitting metaphor since most things users did with a personal computer were things that you could do sitting at a desk: scheduling appointments, writing letters, and creating spreadsheets (see figure above to see what a desktop interface initially looked like). However as soon as the computer turned into a machine that could do more than that, the metaphor turned into an obstacle when it came to helping the user navigate the computer (Norman, 1998). How for instance, would a designer fit a game within the three dimensional representation of a desk? Thus only the initial personal computers with graphical user interfaces stayed true to the metaphor. Shortly thereafter designers realised that the metaphor no longer fitted and abandoned it. The result is that what is now being called a desktop has few similarities left with real desks.


**Figure 3-10: A typical desktop on today's computer.**

In conclusion we can say that while for navigating Information Spaces a metaphor can be useful to help users acquaint themselves quickly with the new environment there are many pitfalls to be taken into account.

### 3.7. Mode

The interaction mode, or sometimes referred to as the *modality,* of an interactive system concerns the interface between the consumer and the system. It specifies both how things are presented to the consumer and how he or she can provide feedback. Schomaker et al. (Schomaker, Nijtmans, and et al, 1995) distinguish between six sensory modalities.

- **Visual**
  Concerned with seeing (compare with *optical*).
- **Auditive**
  Related to the sense of hearing (compare with *acoustical*).
- **Tactile**
  Experienced by the sense of touch.
- **Haptic**
  tactile as perception-modality is distinguished from haptic as output manner.
- **Optics/optical**
  Refers to physical quantities and laws rather than to physiological ones. (Compare with *visual*)
- **Acoustics/acoustical**
  Refers to physical rather than to physiological quantities and laws. (Compare with *auditive*)

These are very broad categories and each *allows* different *interaction* modalities. Interaction modalities are:

- Text
- Graphics
- Audio
- Video

A text and a graphics based interface would both fit under the visual sensory mode, but are distinctly different from an interaction-design point of view. It is therefore important to distinguish between sensory modalities and interaction modalities. The latter has the focus here. The interaction modalities can in turn be divided in sub categories, for instance a 2D and a 3D interface are both graphical modalities but are sufficiently distinct from an interaction-design perspective to justify different treatments.

The mode of an interface can range from simple text interfaces to virtual reality environments. It also covers more unexpected interfaces such as physical objects that users can manipulates to control virtual entities[79] or speech interfaces.

---

[79] Dual augmentation is the term for the interplay between the real and the virtual. See (Mankoff, Somers, and Abowd, 1998) for an example of inspiring research in this field.


**Figure 3-11: A 3D (VR) interface using a geographical metaphor.**

As was the case with the other layers in the hierarchy of navigation one mode doesn't have to exclude the other. It might be necessary to combine modes to achieve an optimal user experience. For example when a user switches tasks during the same session and each task is easier in a different mode. An example might be a system administrator going from drawing a network overview to adjusting system parameters. The drawing might be done using a graphical two-dimensional interface, while she might prefer a terminal window (text mode) for the parameter adjustment. The Linux operating system is an interesting point in case. Many of its users are historically prolific in their use of text interfaces. Nevertheless during the last couple of years various graphical interfaces have been added to the operating system and embraced by many, perhaps even the majority of users. But text remains an important mode for the average Linux user and task context dependent switching between different modes is a "natural" way for them to use a computer.

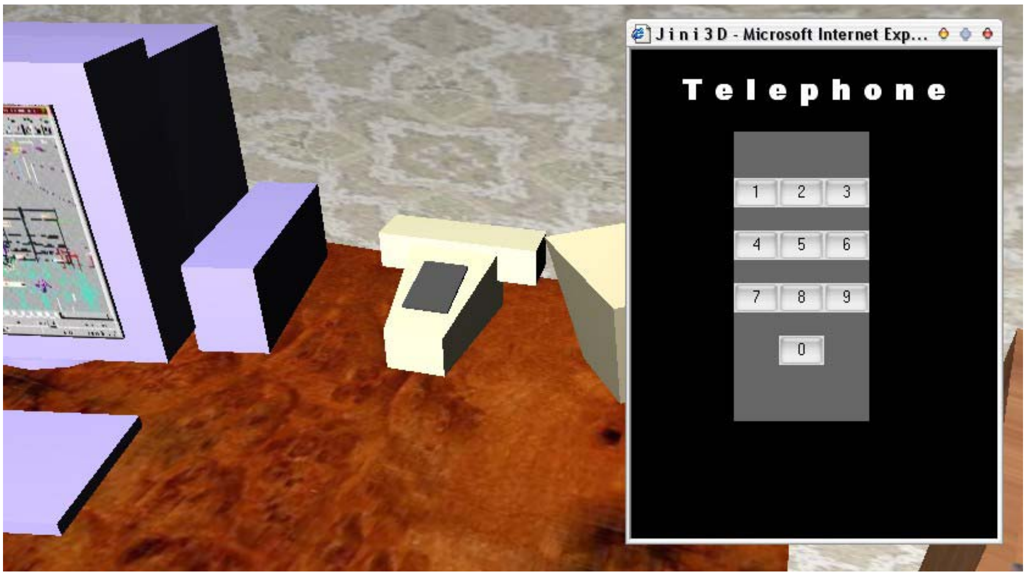



**Figure 3-12: A combination of a 3D and a 2D interface.**

What the appropriate mode is depends upon the situation in which the users find themselves. When sitting behind a desktop computer the users have their hands free, and typically have a large display device at their disposal. Thus they can use a "fancy" graphical mode. On the other hand if they are driving a car they should keep their eyes on the road and hands on the wheel limiting the number of possible modes. A speech interface might be the only solution in that case. The underlying system[80] however might still be the same, as was the case with the 'rooms' project (Henderson and Card, 1986, 211-243) mentioned in section 3.5.2 for which both 2D and 3D interfaces were developed.

While the mode layer sits on top of the metaphor layer in the hierarchy of navigation, it will be recalled that metaphors are optional, thus what is being presented to the user (in text, 2D, sound or any other mode) might be abstract. The often mentioned 3D virtual reality world that uses a geographical metaphor by mimicking a real place looks convincing[81], but if a metaphor was lacking it might look rather abstract, as can be seen in the graphic below:

[80] Being the lower layers in the hierarchy of navigation.

[81] A combination of mode and metaphor that is less common is the auditory rendering of the metaphor of a house, which was done by Lumbreras (Lumbreras, 1993) when researching the use of hypertext for blind people.

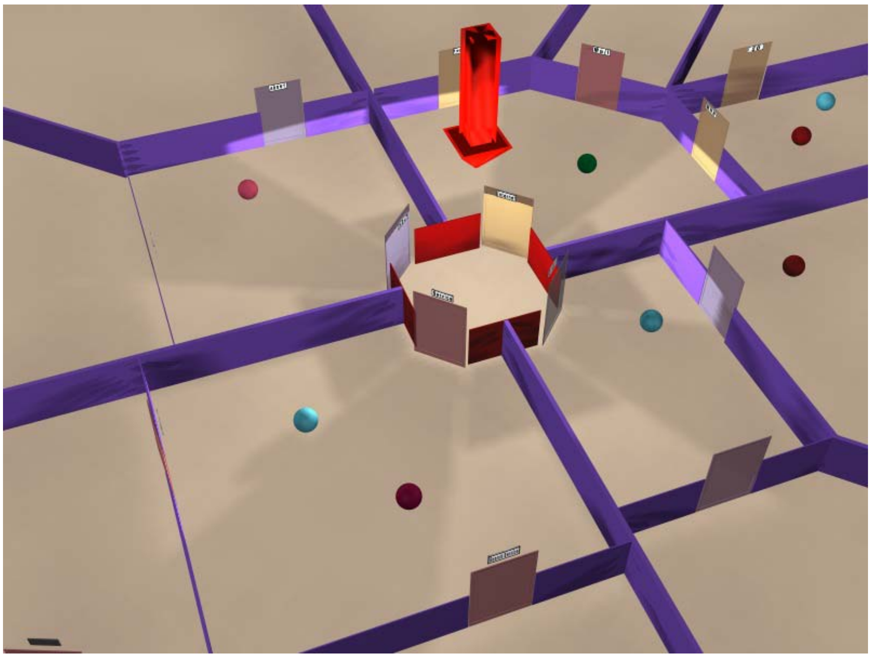


**Figure 3-13: A 3D interface using a topic wise organisation and no metaphor.**

The user starts in the middle under the red arrow. Each door represents a topic, after going through a door the user is presented a new set of doors, each representing a subtopics. This is a recursive process, and by going through the relevant doors the user chooses the topic. The services are represented by 3D icons in each compartment (small spheres in the screenshot).

An abstract interface doesn't mean it is harder to navigate, it only means that it might take longer for users to become accustomed to the environment as they can't use their skills from other domains.

Within each mode a designer has still a lot of freedom. Elastic Windows (Kandogan and Schneiderman, 1997) is an example within the 2D mode[82]. It is an alternative approach to window layout management providing an optimised spatial layout and rapid multi-window operations. When this is compared with the window layout of today's operating systems the possibilities *within* each mode are made clear.

The mode of interaction influences the way users navigate Information Space. It makes a difference whether they use voice commands while driving their cars or

---

[82] See section 3.5.2 for a screenshot of Elastic Windows.

whether they are on a couch wearing virtual reality glasses to navigate information. Which mode is better depends upon the situation, the user, the system, and the task.

## 3.8. Navigation technique

The final layer represents the different ways in which a consumer can navigate among and between the entities. Dourish & Chalmers identify three main navigation techniques in 'Running out of space' (Dourish and Chalmers, 1994): spatial, social, and semantic. In (Munro, Höök, and Benyon, 1999) Dourish gives a good example that illustrates the difference:

> *"Imagine browsing a bookstore. If I pick up a new book because it is sitting on the shelf next to the shelf I've just been examining, then I'm navigating spatially. If I pick up another book because it was referred to in a citation in the first book, then I'm navigating semantically; and if I pick up yet another because it was recommended to me by someone whose opinion I trust, then I am navigating socially"*
> *- P. Dourish in (Munro, Höök, and Benyon, 1999)*

The boundaries between spatial-, social-, and semantic navigation are a guide to distinguishing among the ways the in which a user interacts with the system, rather than between types of systems. They might, and often do, overlap. It is not difficult to imagine semantic- combined with spatial navigation a simple example (in which there is even a touch of social navigation) might be following a road sign in a virtual reality system. By distinguishing amongst these three ways of navigation, and understanding how structure and navigation are linked, designers of information systems can make judgments that better support the user's goals and activities.

As regards the question of why we distinguish between these different ways of navigation it is helpful to quote Dourish & Chalmers:

> *"The most important benefit is a means to understand the value of various interactive models in Information Spaces. We believe it is important to appreciate the ways in which these different navigation techniques are employed, and that this can provide a means to address some problems with purely spatial models"*
> *- (Dourish and Chalmers, 1994)*

As research in this area progresses other techniques of navigation might be identified but here we will discuss the main points of social, semantic, and spatial navigation.

### 3.8.1. Social Navigation

Social navigation encompasses all social aspects involved in navigation. Dourish & Chalmers who coined the term (Dourish and Chalmers, 1994) describe it as

“…navigation through an Information Space that is primarily guided and structured by the activities of others within that same space” (Dourish in (Munro, Höök, and Benyon, 1999)). According to (Forsberg, Hook, and Svenson, 1998) this research field is inspired by the fact that ‘‘that most information navigation in the real world is performed through talking to other people. When we need to find information about an illness, we talk to our relatives, friends and medical doctors, when we are lost in a city, we approach people walking by, etc.’’ While social navigation, just as spatial navigation is not limited to Information Spaces as defined in this thesis that is where this research focuses.

The contribution of the study of social navigation to Human-Computer Interaction (HCI) research is that all interaction with computers can be seen as navigating Information Spaces (Benyon and Höök, 1997;Benyon, 1998, 153-156). Thus while HCI research traditionally saw the user as separate from Information Space, and trying to interact with it, social navigation sees users as a part of Information Space. Moreover just as humans use social methods for navigating the real world, research in social navigation tries to identify the social methods used while navigating Information Space. Social navigation is very broadly focussed, the creation of Information Spaces, the behaviour in them, the social aspects of information creation, users as group members, and the very nature of information itself, are all considered to be topics of social navigation (Munro, Höök, and Benyon, 1999).

Social navigation is commonly subdivided into *direct* and *indirect* social navigation. *Direct Social Navigation* is typically a synchronous process involving direct communication between two individuals. An example of direct social navigation would be my telling a friend where she can find information about concerts in my town. Indirect social navigation is asynchronous and provides information indirectly. It is a by-product of our activities and doesn’t have to be created consciously; an example could footprints in the snow that others might use to find their way.

The current flood of Weblogs gives a good example of how social navigation of Information Space works in the real world. As we saw before[83] a Weblog is a personal Web page with comments on various topics (often information on the Web). Bloggers, (as people with a Weblog are called,) link to Web pages (often other Weblogs) when they think it contains something interesting. At the core this is a system for making recommendations and users can navigate from Web page to Web page using such a system. Many Weblogs are about a specific topic and in turn link to other Weblogs about the same, or a closely related, topic. It offers the opportunity to use Weblogs as filters for the Web and only navigate the information recommended by likeminded persons. This is exactly what social navigation is about.

[83] See section 3.5.

Some researchers such as Richard Boardman (Boardman, 2000) view social navigation as part of a higher-level activity and prefer to talk about *activity-centric social navigation* since that incorporates the *intention* behind the navigation. Whether this is a fruitful approach remains to be seen, but if it is it will be interesting to see whether a new navigation technique can be defined: activity-centric service navigation.

Social navigation is a relatively new look at HCI. Forcing designers to look at navigating Information Spaces as a social activity opens up a wide range of design solutions that might not have been thought of before.

### 3.8.2. Semantic navigation

Semantic navigation means that users moves from one item to another based upon the similarities of meaning, they rely upon the semantic structure of Information Space. An example of this is the Web where Web pages about the same topic are linked *via* hypertext. Looking for a new house on the Web can mean visiting many different Web pages but they are all about the same topic. In this case the user navigates from one Web page to another based upon the semantic relationship between them. Most Web browsing can be classified as semantic navigation.

McKnight, Dillon & Richardson point out that semantic navigation raises an interesting question (McKnight, Dillon, and Richardson, 1993), "...to what extent does a user or reader need to find his way about the argument that an author creates as opposed to, or distinct from, navigating through the structure of the information?" (p.85). Following the thought trail of the author is a complex but necessary task if one wants to extract the semantic relationships in a document. Neumueller (Neumueller, 2001) sees it as a source of navigational difficulties in hypertext:

> *It does not seem surprising that we feel at home only in the hypertexts we have written ourselves, because our memory can only be extended with thoughts we have already had. New ideas have to be created or re-created from texts or conversations. Thus, the [...] reason for our navigational difficulties in hypertext is that, most of the times, we have to follow other people's thought trails until they are our own.*
> *- (Neumueller, 2001)*

An attempt to improve the semantic navigation experience on the Web is the Semantic Web project being developed at the World Wide Web Consortium[84] (W3C). The W3C is working on improving how users (as well as software) can navigate the Web semantically. In words of Berners Lee et al:

---

[84] http://www.w3.org/2001/sw/

*"The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*
*- Tim Berners-Lee, James Hendler, Ora Lassila*

The reason for developing such an extension to the Web is that nowadays most information on the Web is in a format, HTML[85], which makes it hard to use in other ways than for Web browsing. HTML is a way of describing what the content should look like, not what it means. But for software to work with the data and use it in different contexts, it should be able to extract the meaning from the text. As was seen in chapter 2, this can be quite problematic. The solution offered by the Semantic Web is to add a layer of metadata[86] which *describes* content and can be used by humans, agents, and search engines for instance to facilitate navigation.

In chapter 2 Chalmers was quoted (Chalmers, 1999) indicating the problem with metadata for humans. Even if the amount of metadata is just a small fraction of the size of the document described, it will still be a huge amount of metadata when looking at the Web at large. It might even require meta-metadata, thereby bringing users back to where they started. For machines with their capacity for text manipulation metadata makes the information easier to process.

The question remains how of how faithfully metadata describes data. The limitations of language to do so were pointed out in chapter 2.

### 3.8.3. Spatial navigation

Navigation through space means that a sense of metric distance is being used in going from one item to another. The user can move *above, below, outside*. A well-known example from the area of Information Spaces is a virtual reality world where users have to bridge a virtual distance, expressed in the virtual metric, to go from one item to another. Spatial navigation is a *technique* of *navigation*, which shouldn't be confused with the *organisation* of information. Here the focus is on the user while when talking about organisation it is upon the environment. (Which is why theories discussed in section 3.5.4 are mainly from environmental design (e.g. city planning, architecture) while theories addressed here are mainly from psychology (e.g. cognitive science)).

---

[85] Hypertext Markup Language, the document format on the Web. Webpages are written in HTML so Web browsers can display them.

[86] Metadata is descriptive information about data (Tweedie, 1997).

Thinking in spatial terms is a fundamental trait of human cognition[87], and can for example be seen in the way in which users talk about navigating the Web. They 'get lost', 'go' to a certain site, 'go back' to a site, 'get on the Web'. It indicates that humans use spatial concepts when thinking about navigation. How successful someone is in spatial navigating depends upon two factors, spatial ability and spatial knowledge. Spatial *abilities* are cognitive functions that enable people to deal effectively with spatial relations, visual spatial tasks and the orientation of objects in space. This has been researched extensively for many years within psychology (see (James, 1890) for an early example). Spatial *knowledge* on the other hand is concerned with knowledge about the *environment itself*. It includes sensory information (e.g. what does the place look/smell/feel like?) as well as information about distances and directions from place to place. Furthermore it includes inferred knowledge about places not yet seen and paths not yet travelled. Thorndyke (Thorndyke, 1980) distinguishes between three hierarchical types of spatial knowledge:

1. **Landmark knowledge**
   Information about the visual details of specific locations in the environment. It includes memory for notable perceptual features such as a unique building.
2. **Procedural knowledge**
   Information about the sequence of actions required to follow a particular route. Procedural knowledge is built by connecting isolated bits of landmark knowledge into larger, more complex structures.
3. **Survey knowledge**
   Configuration or topological information. Object locations and inter-object distances are encoded in terms of a geocentric, fixed, frame of reference. A geocentric frame of reference is a global, map-like view while an egocentric frame of reference is a first person, ground-view relative to the observer.

When dealing with Information Spaces spatial navigation only makes sense when the information objects are spatially laid out, as in the virtual reality world previously mentioned. When it is not, such as on the Web, spatial navigation is not possible. As humans excel in spatial navigation there have been several attempts to make the structure of the Web explicit by superimposing a spatial structure, which could facilitate navigation (see (Dieberger, 1995) for an example). With an explicit structure users could develop their spatial knowledge about Information Space and improve their navigation. Some attempts to make the structure of hypertext systems, that "tends to hide this structure" (Dieberger, 1995), more explicit have had some success (Dieberger, 1995).

[87] Steven Pinker's 'How the mind works' is an excellent introduction to the field of cognitive science, in the broadest sense of the word, and it discusses this topic thoroughly (Pinker, 1997).

To conclude the discussion of navigation techniques in this section, it is clear that while describing them as completely separate entities is useful, it is not very realistic. It is useful to identify the different properties of each technique, but problems arise because often the different ways of navigation are combined. One common combination is between *semantic* and *spatial* navigation. Imagine for example an Information Space where information objects are grouped according to semantic relationships, but are rendered to the user as spatial dimensions (such as in the example of a folder hierarchy in section 3.5.4). In such a case semantic relations are mapped onto a spatial arrangement. Navigating this space would consist of simultaneously spatial and semantic navigation. And if one were to add navigation following another actor's advice there would even be a social navigation component.

## 3.9. Summary

This chapter discussed the theoretical background of navigation. To structure the discussion a hierarchy of navigation was proposed that ordered all relevant aspects of navigating Information Spaces. Subsequently each topic was treated in more detail. First theories of navigation in general were addressed. Since much of the research is based upon those theories it is important to understand their most important assumptions. Before continuing with the navigation of Information Spaces the relevant concepts had to be defined, so once *navigation* and *Information Space* were defined the hierarchy of navigation was developed. The bottom layer concerns the *entity* that is being navigated, which can be plain information (documents), functionality (services) or a combination thereof. The next layer in the hierarchy specifies how the provider or consumer *organises* these entities. A few examples were worked out: time-based, activity-based, spatial and semantic organisation. The type of organisation affords different *metaphors*, which is the next layer in the hierarchy. An activity-based organisation of documents can for instance use a calendar metaphor. The mode layer in turn specifies how it is all rendered to the user. There are many *modes* possible, for example a speech, 2D, 3D or a haptic mode. Finally the top layer, *navigation technique,* is about the way the user navigates the entities, possibilities are spatial-, semantic- and social navigation. The difference between the top layer and the underlying layers is that it is about the user instead of Information Space, making that the theories are mostly from psychology.

The next chapter will focus on an Information Space that not only consists of documents but also of services and see whether that makes a difference in terms of navigation. The chapter provides definition of what an Information Space consisting of documents *and* services actually is, and what the prerequisites are for such a space to become a reality. Subsequently both Information Spaces consisting of documents and Information Spaces consisting of documents & services will be compared to see what the differences are and whether those differences have an effect on the navigation of both spaces. Finally the chapter will discuss whether, and how, the state-of-the-art addresses any newfound problems.

# 4. Navigating Distributed Services

*This chapter will investigate which parts of the discussion in the previous chapter relied explicitly on documents. It will explain in detail what is meant by 'navigating distributed services' and provide the relevant definitions as well as an overview of the state of the art in the industry at the time of writing. The final section will identify issues that have not yet been addressed by the state of the art.*

The scenarios developed in chapter 2 explored the broad consequences of the three changes mentioned in the introduction chapter and the effect upon navigation was chosen as the core topic for this thesis. Now that the existing theories on the navigation of Information Spaces have been explored in the previous chapter it is time to look at virtual spaces consisting of services and discuss navigation of such spaces. To limit the thesis' scope the focus will primarily be upon the effect of introducing distributed services, and to a lesser extent on the effect of the increase in ways of accessing such services and information. (The latter will be discussed in this chapter where relevant).

It is necessary that I first explain what a service constitutes and why it is a fitting terminology in the context of this thesis. Following that an Information Space as defined in the previous chapter will be compared to a virtual space that consists of services. This will in turn be followed by a discussion of the prerequisites for such a space to become a reality. How a space consisting of services can be navigated is the topic of the subsequent section, followed by an overview of the existing attempts in both industry and academia.

## 4.1. Services

It is expedient at this point to repeat[88] Waldo's (Waldo, 2001) description of the services perspective.

> *"Services are an idea that says what you go looking for is not a machine or a particular thing, but something that provides a service for you. Essentially, it is a way of doing late binding -- you don't go out looking for the particular thing that you know does this for you. [Instead] you go looking for whatever it is that you want done."*

To limit the research domain to *digital* services that are *accessible via the network* the working definition used here is:

[88] See section 1.1.2 for an introduction to services.

*__Service:__*
*The offering to solve a task made available through a digital network.*

There are three reasons why the term 'service' is more fitting than 'application' when talking about functionality available on a computer. The first is that it more precisely describes what is being offered. It forces one to think about functionality instead of specific implementations offering it. Thus instead of talking about the applications Emacs and Vi we should, according to the services perspective, talk about "Text Editing Services."

The second reason has to do with the fact that one application can offer multiple services; it can do more than one thing for others. This is not made explicit when using the term 'application'. A photo editing application for instance not only offers the service of editing images but might also offer finer grained services such as adjusting the brightness or the size of an image separately. Third parties might offer some of these sub-services again. An aggregation of services that looks like one service to the user represented schematically could look like this:

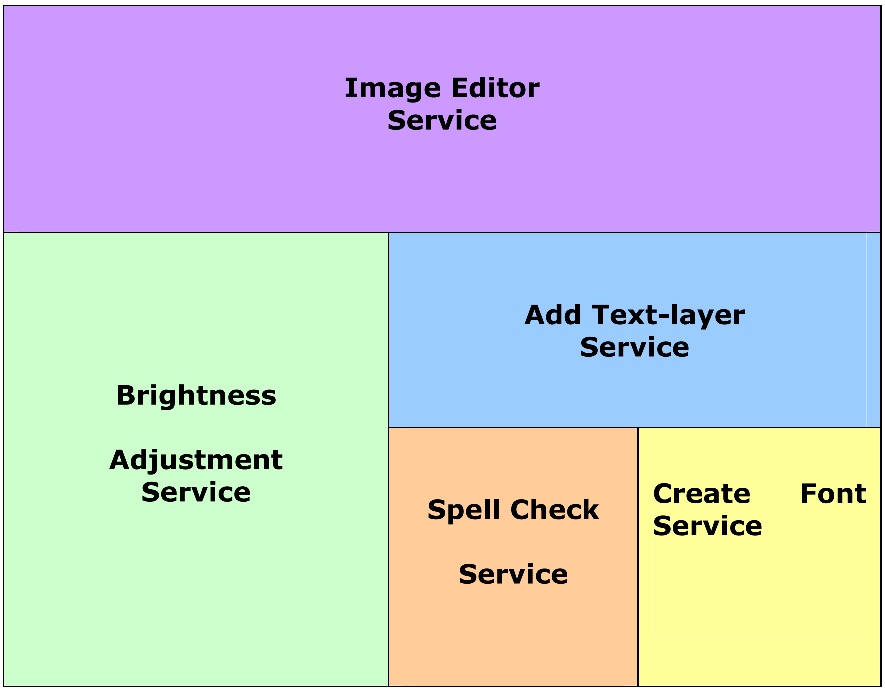



**Figure 4-1: One service built out of multiple sub-services.**

If users want to first adjust the brightness of an image and then do some spell checking on the text layer they might use the same application twice without realising. They can first search for an 'Image Brightness Adjusting Service' and then for a 'Spell Check

Service'. If what was available was described in terms of applications they would have had to find out themselves, which applications offered what they wanted to have done.

The third reason to use the term 'service' is that, for our purposes, the term 'application' is historically too closely connected to computer programs running on standalone desktop computers. This could be limiting, for instance when describing functionality that is distributed across a network or offered by a device. 'Service' covers the whole range. An example would be a television that is connected to the network. It would be difficult, and confusing, to describe the functionality it offers to the rest of the network in terms of 'applications', but when using 'services' one could say that the television offers a Display Service, a Sound Service and so on.

As this thesis' primary focus is upon the functionality offered *via* the network it makes sense to discuss services rather than applications. Fortunately it is possible to describe an application in terms of the service(s) it offers, thus existing applications[89] are not excluded by my choice of terminology.

## 4.2. Services Sphere

In the previous chapter *Information Space* was defined as "the information available at a given location through digital media, as perceived by a user". The focus was upon information, rather than services. Although the 'entity' layer in the hierarchy of navigation mentioned 'services' the rest of the chapter focused upon the existing body of work on navigating information[90] (documents). This should not be surprising, as most of the research on navigating Information Spaces has dealt with virtual spaces consisting only of information. As was pointed out in the introduction chapter it was not until recently that services were considered to be a part of the virtual spaces. Prior to that Information Space could be represented diagrammatically as interconnected documents:

[89] Except applications that don't make their services available through the network, they *are* excluded based on the working definition of a service.

[90] See section 3.3.

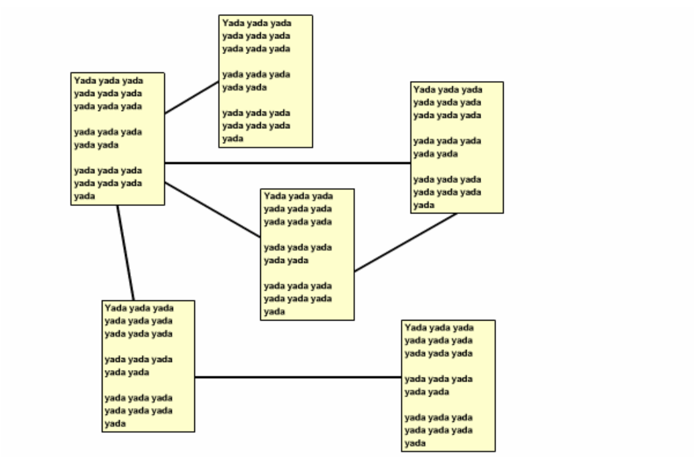



**Figure 4-2: A document centred Information Space.**

The picture changes when services are introduced into Information Space. What is accessible is no longer merely plain information but also services. This change affects Information Space in two ways. Firstly services differ widely from documents and each other in their capabilities. This was illustrated in the scenarios in chapter 2 which pointed out that a service can consist of anything from a simple calculator service to a complex service capable of monitoring all of the devices in a hospital. Clearly the difference in usage between two such services is greater than that between reading two documents be they never so complex. Secondly it is possible for services to use each other, a simple example is a text editor service making use of a spellchecker service. Thus where the definition of Information Space in the previous chapter[91] mentioned only interconnected documents, the new definition should incorporate interconnected services as well. This is not to exclude documents — manifestly there will always be documents of some form or other, thus a diagrammatic representation of Information Space that has been extended with services could look something like the graphic below:

[91] "The information available at a given location through digital media, as perceived by a user".

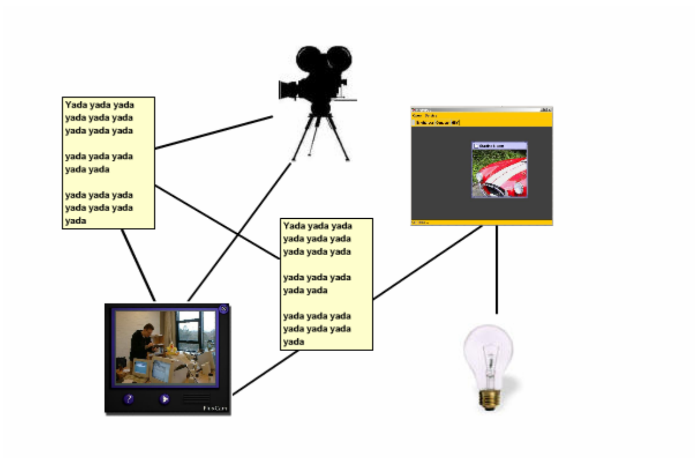


**Figure 4-3: A services centred Information Space (Services Sphere).**

The extended Information Space consists of interconnected services and documents, which differ markedly. I submit that it shows that the name *Information Space* has become inappropriate, I therefore propose that the name *Services Sphere* is appropriate for Information Space(s) augmented by services. In the interests of clarity however I must enter a caveat at this point. In order to keep the working field of this research manageable I must limit the scope of the working definition to 'services'. Therefore I replace 'information' in my working definition of *Information Space* (see section 3.2) with 'services'.

> ***Services Sphere****:*
> *The services available at a given location through digital media, as perceived by a user.*

This working definition doesn't include information so a Services Sphere only consists of services, not documents. Follow-up research can include documents again.

## 4.3. Prerequisites for these to be relevant questions

As well as the caveat entered above it is necessary to draw attention to the changes that have to take place before a Services Sphere comes into existence and navigating it will become a real topic. Doing so will enable our continuing to discuss the effects upon navigation of the transition from Information Space to a Services Sphere. The main prerequisite is that using services distributed *via* the network should be as common as reading documents distributed *via* the network is now. Implicit within this is that the amount of services available *via* the network needs to increase dramatically. If there were only a few services available the problems with navigating them are less than when they are as ubiquitous as documents are today. What obstacles have to be overcome for this to become a reality?

Manifestly the most important prerequisite is that services will have to be made available *via* the network. Without such services there won't be a Services Sphere. It is equally apparent that for such services to come into being, some conditions, both technological and non-technological, need to be met.

## Technical prerequisites

At present there are still many technological gaps that need to be filled. The problems can be split up according to whether they have to do with supplying, transporting or accessing services.

Firstly there are the technical problems inherent to *supplying* services. Consider the following: a company wants to supply a text editing service, which prevents users from having to install applications on their own computers. The company will need to make the service available *via* the Internet. To do so it will need to run the service on one of it's own servers and make it available in such a way that a remote user can access and use it. This requires different technologies than are needed for applications for the personal computer, which required the company to distribute the software on physical carriers. It may also require the company to completely rewrite their application to make it available as a service. Because of the different nature of networked services most existing applications will need at the very least *some* reworking. That is because the networking aspect of service provisioning introduces new constraints. At present the network is:

- Unreliable,
- Insecure,
- Dynamic.

Moreover

- it introduces latencies,
- there are costs associated with data transfer,
- and there are bandwidth limitations.

All of these are matters that a service provider needs to take into account when migrating from providing user-installed local applications to the provision of distributed services.

As has been mentioned before, it will be not just software companies providing services, some services could equally well be provided by devices. An example of this could be a VCR[92] offering the service of recording television programs. There are few,

---

[92] Video Cassette Recorder.

if any, VCR's capable of publishing their services on the network chiefly because such a change would require many hardware changes. Most devices that offer some kind of service that could, in theory, be made available *via* the network are not yet equipped with the required hardware and software.

Secondly there are the technical problems of *transporting* data or functionality either between services or between parts of a service. The difference between a service-oriented network and a document-oriented network is that functionality is 'shipped around' rather than merely plain data. Besides that there *might* be more communication between services or parts of a service. In case of the text editor example part of the functionality might be executed on the service provider side (for instance a relatively CPU[93] intensive job such as calculating the number of nouns in a document). The difference in what and how much is transported entails differences in communication between the consumer and provider of the service compared to a document-oriented network. This doesn't mean the bandwidth requirements are necessarily going to be higher, but the effects of network security, failure and latency might become more important.

Thirdly there are the technical problems inherent in *accessing* services. The different interaction characteristics of services compared to documents (Nielsen, 2000a) means that accessing services has different requirements as well. To give an example, some services, such as the mentioned programming of a VCR from a remote location requires more user input than reading a document. Such input could be as dissimilar as speech and handwriting, making the possible differences in access even greater. Another difference is that some services might only be useable in certain situations, such as a service for measuring atmospheric humidity, requiring them to be compatible with devices that can be used in those situations.

## Non-technical prerequisites

The non-technical prerequisites can be divided into obstacles for services providers and for their consumers. For the providers of services the business processes around a service needs to be adapted to the services paradigm, not only the technology. Thus for instance a software company would have to set up a billing mechanism that charges users on a per use basis for instance rather than selling CD-ROM's with the software and then (possibly) charging for downloadable upgrades. This in turn affects the marketing of the product, the service contracts, the lifecycle of the product, how to deal with any possible competitors and many other aspects that are far beyond the scope of this research.

Whether services will proliferate also depends on the demand side. Without any demand the Services Sphere will never come into being. Consumers will have to grow

[93] Central Processing Unit.

accustomed both to wanting and using services. Since a push-market seldom works[94] a demand for services is needed. Once an adequate demand generates the first services, it will bring more services in its wake. This is a chicken/egg situation since an adequate supply of services will stimulate the demand for more. There are several aspects to stimulating demand. Each service should offer the right combination of utility, accessibility and usability.

> **Utility** — Where a service or services provide something that can't be provided otherwise it can act as a catalyst.
>
> **Accessibility** — The service has to be accessible, which means that the user can use it when she wants; at the right time at the right place, in the way she prefers.
>
> **Usability** — The service has to be usable. Although functionality can compensate for usability[95] it can never substitute for it entirely.

Everything being taken into account there are many problems that need to be taken care of before a Services Sphere can become a reality and navigating it becomes a real issue. However the number of problems solved is slowly increasing[96], which has been the motivation behind this research. Section 4.5 will show that there are already quite a few commercial efforts to solve the problems. This begs the question of why, since such efforts require a large investment by all parties. This is hard to answer since most of these attempts barely left the research laboratories and most investors tend to be rather secretive on this matter. Partly this is because the expectations[97], and thus the risk, regarding network delivered services are high. Whether this speeds up or slows down the transition from Information Spaces to Services Spheres remains to be seen.

## 4.4. Navigating the Services Sphere

Using scenarios chapter 2 broadly identified the three research areas for this thesis;

- the differences between navigating services and navigating documents,
- the organisation of services and the use of metaphors,
- metadata for services.

---

[94] The initial failure of WAP phones in Europe is a convincing example.

[95] The success of Short Message Service (SMS) in Europe during the last five years is a good example. Typing a text message using the small numeric keypad of a mobile phone is far from user friendly, but the utility of sending short text messages outweighed the usability disadvantages judging by the success of the service.

[96] See the introduction chapter.

[97] The current hype surrounding Web services is a good indicator.

After discussing the theoretical backgrounds on navigation in chapter 3 and the Services Sphere in this chapter it is time to develop our exploration of the topics raised in more detail.

The bottom layer of the hierarchy of navigation[98] makes a distinction between whether it is documents or services that is being navigated. Inter-entity differences influence navigation. For example, are users looking for a document to work on or are they looking for a service that will operate upon a document? At present, using a desktop computer and local applications and documents, it is quite common to use both strategies. There is still much research going on in the area of Human-Computer Interaction to find out what makes most sense in different situations[99]. Since there is no definite answer to that question we will see both strategies in navigating Information Spaces.

The differences between navigating information and navigating services was noted by Web usability expert Nielsen in one of his newsletters called '*The network is the user experience: Microsoft's .Net announcement*' (Nielsen, 2000a). In the newsletter he mentions, "Hypertext theory has predicted the emergence of a navigation layer that would be the nexus of the user experience". He goes on to admit he always assumed that new nexus would be the convergence of the Web browser with the operating system, but he now realizes that the browser is not fit for offering 'information objects' since "application functionality requires more UI than document browsing" (Nielsen, 2000a). His distinction between 'information objects' and 'functionality objects' fits with the assumptions this thesis is based on[100].

Combining the concept of the Services Sphere with the theories on navigation discussed in the previous chapter raises the question of what the differences are between how users navigate documents and how they navigate services. One question in this regard is whether the theories of navigation that were used for navigating Information Spaces are also applicable to discussions on Services Sphere navigation. For instance the guidelines from Kevin Lynch for constructing easy navigable cities (Lynch, 1960) were used for constructing Information Spaces (Darken, 1996;Dieberger, 1994) and the question arises of whether they are also of use when constructing easy navigable Services Spheres. Clearly this can only be the case if Lynch's findings are translatable to the realm of navigating services.

---

[98] See section 3.3.

[99] See section 1.3 for an overview of research projects working on these questions.

[100] See the introduction chapter for the three changes that form the basis of the assumptions. The quote furthermore shows that Nielsen has his doubts whether the new navigation layer could be built using existing technologies, a topic close to the heart of this thesis.

Benyon & Höök's definition of navigation in the context of Information Space (Benyon and Höök, 1997) consisted of three types of activity; way finding, exploring and object identification. The question therefore is whether the same activities can be identified when navigating services.

When looking closer at the hierarchy of navigation two layers can be identified that might differ whether the entities being navigated are documents or services; the organisation and the metaphor layers. These layers are closely connected since a choice in one layer limits the choice in the other. Chapter 3 identified several ways of organising information (time-, activity-, semantics-based) and several metaphors that have been tried out and studied (desktop, geography). The question is whether the same means of organisation and the same metaphors are usable when talking about navigating services.

The last questions to be raised deal with metadata for services and its connection with navigation. The scenarios highlighted the need for metadata with services and chapter 3 discussed the role of metadata in navigating Information Spaces. A semantic organisation of the Web was discussed, and the Semantic Web project (McIlraith and others, 2001, 46-53;Berners-Lee, Hendler, and Lassila, 1 A.D.) is a good example which relies heavily upon the use of metadata. This raises the question if, and what, metadata is needed from *services* to ease navigation.

The background literature study hinted that as of yet there is no clear answer to these questions. The next section will explore whether this was correct by taking a look at the existing attempts both in industry and academia to come up with answers. If not research will have to be put in place to come by them.

## 4.5. State of the art

Given that the services[101] perspective is relatively new[102] it is possible to give a good overview of the existing projects. Such an overview is merely a 'snapshot' of the current situation and will be incomplete the moment this research is published. Changes in this field follow each other so rapidly that it makes sense to only compare the different projects on their similarities. These similar aspects are the research topics mentioned in the previous section. I will give a short description of each project followed by an analysis of how they try to answer the questions regarding navigation of services, organisation and metaphors, and metadata.

[101] As defined in section 1.1.2.

[102] Concluding by the small number of (partly) relevant projects mentioned in the introduction chapter and the lack of literature identified by the background literature study.

### 4.5.1. ASP

Application Service Providers (ASP) were the first[103] to provide services *via* the Internet (or sometimes a private network). An ASP is an organisation that hosts software applications on its own servers, its customers rent *the use of* the application and access it *via* the Internet. The most common features of an ASP are (Brain, 2002):

- The ASP owns and operates a software application.
- The ASP owns, operates and maintains the servers that run the application. The ASP also employs the people needed to maintain the application.
- The ASP makes the application available to customers everywhere *via* the Internet.
- The ASP bills for the application either on a per-use basis or on a monthly/annual fee basis.

Compared to locally installed applications, application service provisioning is a logical step towards a service-oriented network since it is not much more than running applications on one server and giving customers access to them *via* the network. The first applications provided in this way were mainly business productivity applications that were too expensive for small businesses to run themselves. Examples are large software packages such as Oracle (database management), SAP (enterprise resource planning) and PeopleSoft (enterprise-wide applications for client/server environments). Later simpler, and more common, software packages were made available in ASP versions as well, for example word processors and spreadsheets[104]. Before applications were available in an ASP version a company had to buy the application and hire personnel to administer both hardware and software.

How users navigate ASP applications depends upon the applications themselves. Some can be accessed using a regular Web browser while others require the client to install an application[105] or plug-in[106]. In case of the Web browser the client finds the ASP application just as she would find a Web page, by following hyperlinks. In the case of a client side application the user launches that application which establishes a connection with the ASP application.

---

[103] Ross Perot, who later became famous for his attempts to become the American president, was during the beginnings of the 1960's the first to experiment with the ASP model. While being an IBM mainframe salesman he leased unused mainframe computer cycles and application space from his bigger customers and sell it as outsourced data processing to mid-sized companies. The difference with today's ASPs (sometimes called Internet ASPs) is that the provisioning wasn't through the Internet.

[104] See for example products from Curl (http://www.curl.com).

[105] Which is the solution Citrix (http://www.citrix.com) for instance chooses.

[106] Which is the case with Curl (http://www.curl.com).

Navigating ASP applications has not been problematic. Since the number of ASP applications is still relatively small (they are not (yet) very common) the chances that an average computer user can't find the application they are looking for is rather small. Current ASPs assumes that the client knows the Web address of the application or has the right client side application installed. An overall mechanism for organising large collections of ASP applications is thus not yet needed. However, if the research corporation IDC[107] is right this might change within the near future (International Data Corp., 2002).

### 4.5.2. Web Services

*Web Services* are the service-oriented environment that has recently received the most attention in the press[108]. Although a standard definition is currently lacking, IBM's Web Services Architecture team provides a useful summary.

> *Web Services are self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the World Wide Web.*
>
> *Web services perform functions, which can be anything from simple requests to complicated business processes...Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.*
> *- (IBM Web Services Architecture Team, 2000)*

The fact that the term *Web Services* came into use at the same time as I started this thesis and is now an important technology for most of the major IT companies (Wong, Kane, and Ricciuti, 2001) is in itself a significant statement about the dynamics of the field. Whether Web Services will be successful depends not only on the supply side of the chain, but equally (or perhaps even more) upon the demand side. Thus far demand is less than the vendors would have you believe, but is slowly increasing (Wong, Kane, and Ricciuti, 2001).

Notwithstanding the lack of a generally accepted definition one might more generally speaking also call Web Services inter-application messaging *via* the Web. At the time of writing there is some agreement on how such communication should be implemented[109], but most of the largest vendors use different technologies to implement Web Services (Wong, Kane, and Ricciuti, 2001). Even though Web

---

[107] International Data Corporation. http://www.idc.com

[108] A good overview report on Web services is (Wong, Kane, and Ricciuti, 2001).

[109] The Web Services Interoperability organisation (WS-I) promotes Web services interoperability across platforms, operating systems, and programming languages as well as across industry and standards organisations (http://www.ws-i.org/).

Services are still in their infancy there is already a large variety of technologies to choose from to implement them[110].

According to (IBM Web Services Architecture Team, 2000) the difference between Web Services and distributed computing is that the latter "requires too much agreement and shared context among business systems from different organisations to be reliable for open, low-overhead Business-to-Business e-business". The rise of Web Services is part of a trend of movement away from tightly coupled monolithic systems and towards more loosely coupled systems consisting of dynamically bound components (IBM Web Services Architecture Team, 2000).

As with any service-oriented environment a few essential actions have to be taken before a service is available to users. A Web Service needs to be:

- **Created**, and its interfaces and methods of invocation must be defined.
- **Published** to one or more Internet repositories for potential users to locate.
- **Locatable** to be invocable by potential users.
- **Invocable** to be of any benefit.
- **Unpublished** when it is no longer available or needed (optional).

Discussing the details of each step is beyond the scope of this thesis[111], but a high-level overview shows that there are three parties involved.

- A *service provider* who creates the Web Service and *publishes* it in a central registry of Web Services.
- A *service broker* who maintains a registry of Web Services (called an UDDI registry[112]).
- A service requester who looks up (*finds)* the Web Service in the registry and *binds* directly with the service provider when a Web Service has been found.

[110] Examples are IBM's WebSphere, Microsoft's .Net framework and SUN's J2EE framework (see (Wong, Kane, and Ricciuti, 2001) for more examples).

[111] And would confuse the reader with an overload of acronyms. The brave at heart can go here to get the idea: http://www.xml.com/pub/a/2002/01/09/soap.html

[112] There are already many public Web Services directories online, examples are provided by companies like IBM, NTT, SAP, Microsoft and Hewlett-Packard. An overview can be found here: http://www.uddi.org/

The diagram below depicts the processes involved:

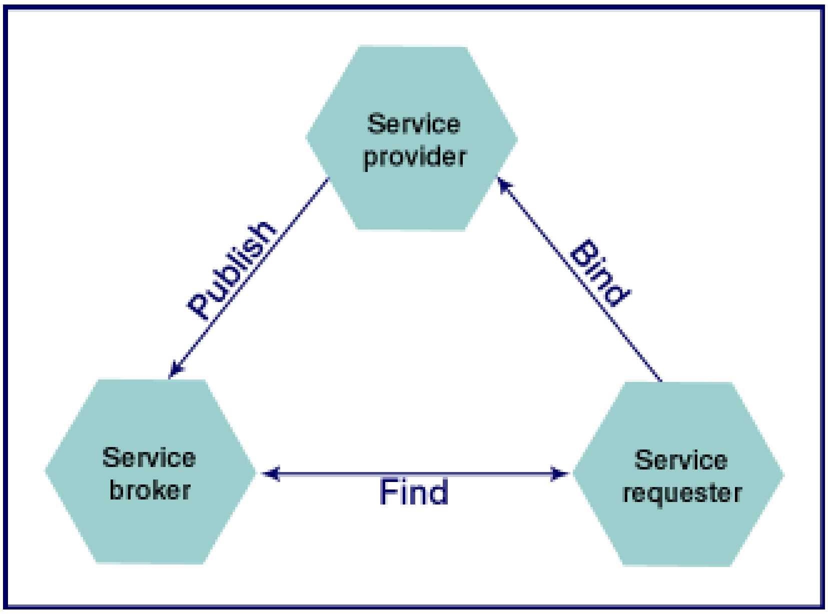



**Figure 4-4: Publishing, binding and finding Web Services.**

The finding of, and binding to a Web Service are actions that can be automated, so the *service requestor* can both be a human or an application. At the moment most Web Services are meant for other applications or Web Services[113]. The reason Web Services are part of this overview of technologies that offer services to humans is that some Web Services are directly aimed at humans. IBM's Web Services Browser (IBM, 2002) allows humans to navigate a directory of human-centred Web Services. (See screenshot below.)

[113] A look at one of the Web Services registries (UDDI registries) will make that clear: http://www.uddi.org/

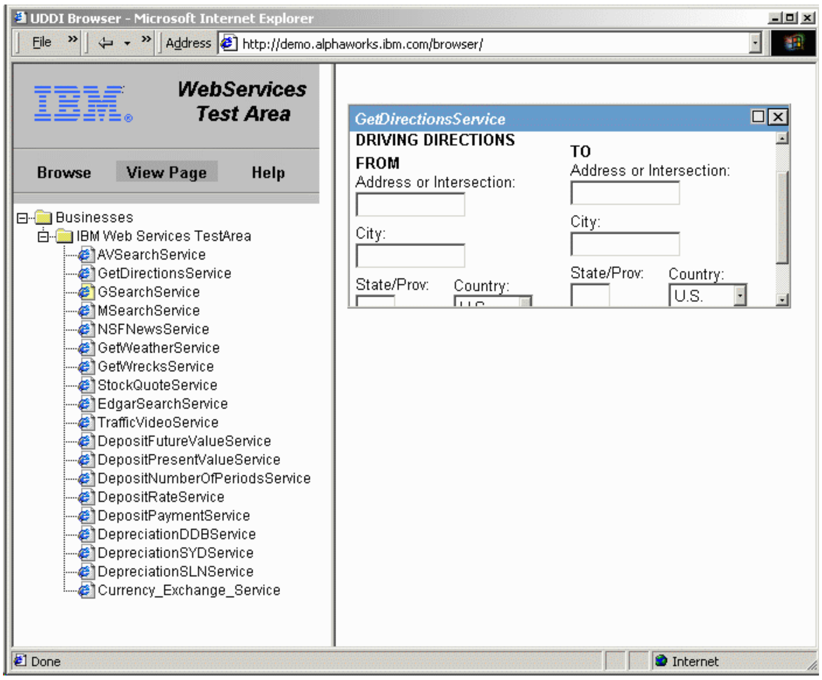



**Figure 4-5: IBM's Web Services browser (IBM, 2002).**

As can be seen in the screenshot the Web Services are organised by service provider ('businesses') and displayed as a simple tree structure. The organisation is based upon the information about the service's metadata, as described in two documents that come with each Web Service and that is stored with the service broker. First there is the Well-Defined Service document containing general information, such as service category, service description, expiration date, as well as standard business contact information about the service provider. Secondly there is the Network-Accessible Service Specification Language document, which describes operational information about the service, such as service interface[114], implementation details and access protocol(s). As yet there is not a standard for describing a Web Service, only the syntax (called Web Service Description Language) has been defined.

Navigation is done by traversing the tree or using a search tool (not visible in the screenshot). This way of navigation is not at present problematic given that there are relatively few Web Services, but with thousands of such services, or even (if they become successful) millions of them, the problems become identical to those posed by millions of Web pages organised in directories or searched using search engines[115].

---

[114] A programmatic interface describes what functions a piece of software provides.

[115] See section 3.5.

Thus far IBM's Web Services Browser is the only human-oriented way of navigating Web Services, but with the increase in Web Services for humans more can be expected.

The Web Services movement still lacks a critical component to make it a success with human users, and that is the software to deliver Web Services in a rich and interactive way. Existing client-side technologies (e.g. HTML or JavaScript) are too limited for building user interfaces for Web Services that meet the high demands of interaction design. One attempt to solve this problem is Curl (Curl cooperation, 2002). Curl offers a specific language for creating rich interfaces and a runtime environment that the clients need to run them. At the moment the runtime is only available for desktop computer operating systems with a Web browser, preventing the use of Curl on other devices.

### 4.5.3. Cooltown

CoolTown is an infrastructure that supports context-aware applications by representing each real world object, including people, places and devices, with a Web page (Caswell and Debaty, 2002). The research laboratories at Hewlett-Packard are developing CoolTown. Kindberg et al (Kindberg, Barton, and Jeff Morgan, 2000) best explain how CoolTown works:

> *"We put web servers into things like printers and put information into web servers about things like artwork; we group physically related things into places embodied in web servers. Using URLs for addressing, physical URL beaconing and sensing of URLs for discovery, and localised web servers for directories, we can create a location-aware but ubiquitous systems to support nomadic users. On top of this infrastructure we can leverage the Internet connectivity to support communications services. Web presence bridges the World Wide Web and the physical world we inhabit, providing a model for supporting nomadic users without a central control point".*
> *- (Kindberg, Barton, and Jeff Morgan, 2000)*

The overall aim of CoolTown is to create virtual bridges between mobile users, physical entities, and electronic services. The motivation behind CoolTown is that, as Kindberg et al put it, "we think the physical world and the virtual world would both be richer if they were more closely linked" (Kindberg, Barton, and Jeff Morgan, 2000).

Navigation takes a central role in CoolTown by providing a (nomadic) user with relevant (context-aware) information and services. A scenario given by (Kindberg, Barton, and Jeff Morgan, 2000) highlights the navigation aspects. Imagine a conference room where all objects contain a Web server serving a Web page that can be updated dynamically. Objects each have a unique RFID-tag[116] that can be used to get information about the object from the Web. Even the conference room has its own Web server that provides a 'PlaceManager', which organizes the Web representations of the objects in the room into collections (much like Web directories). The persons in the room have a Web presence in the form of a personal Web page extended with 'WebLinks'. WebLinks are services to facilitate communications between individuals. When users enter the room with a device equipped with sensors for the RFID tags they can get *information* on all objects and persons in the room, as well as access the virtual representation of the room. The device also allows them to access *services* that are relevant in the physical context, such as a printer or a projector, through the PlaceManager of the conference room.

Navigation in CoolTown for the most part means navigating a physical environment augmented by digital services accessible through their Web presence. As a user equipped with an RFID-tag sensitive device moves from place to place the available information and services change. For this reason much of the work on CoolTown is closely connected to the research field of ubiquitous/pervasive computing[117]. For instance only when a user is in one place and navigates the information and services available are there similarities with navigating Information Spaces. This is because in such a case the situation changes from navigating a *physical space* while Information Space/Services Sphere adapts, to standing still and navigating a virtual space. In the latter situation the user is adapting to (navigating) Information Space /Services Sphere, which is the focus of this thesis.

### 4.5.4. CyberDesk

CyberDesk is an active browser metaphor that can act as an alternative to the desktop metaphor. It attempts to automatically integrate services based upon the virtual context of the user (Dey, Abowd, and Wood, 1998). The virtual context consists of the collection of information and services directly available to the user[118]. CyberDesk looks like (or could look like) a regular 'desktop' but the intelligence in CyberDesk's user interface comes from applications automatically providing their services to the user. Rather than displaying all possible services to the user at all times, the interface is limited to those services that are relevant to the user's context[119]. A service can be either an action that an application can perform, *or* data that an application can

---

[116] Radio Frequency Identification. An electronic label that can be read out by special RFID readers.

[117] See chapter 1

[118] In other words, Information Space/Services Sphere.

[119] Similar to the context menus in current operating systems, but in this case the menu lists available services that can act upon the entity that was clicked on.

provide. These services can come either from locally installed applications or from the Internet.

For example if a user were to select some text in an e-mail CyberDesk would automatically offer several services capable of acting upon the selected text. If the text were a hyperlink it could offer to open the Web page in a browser, search for pages pointing to this Web page and so on. It could look something like this:

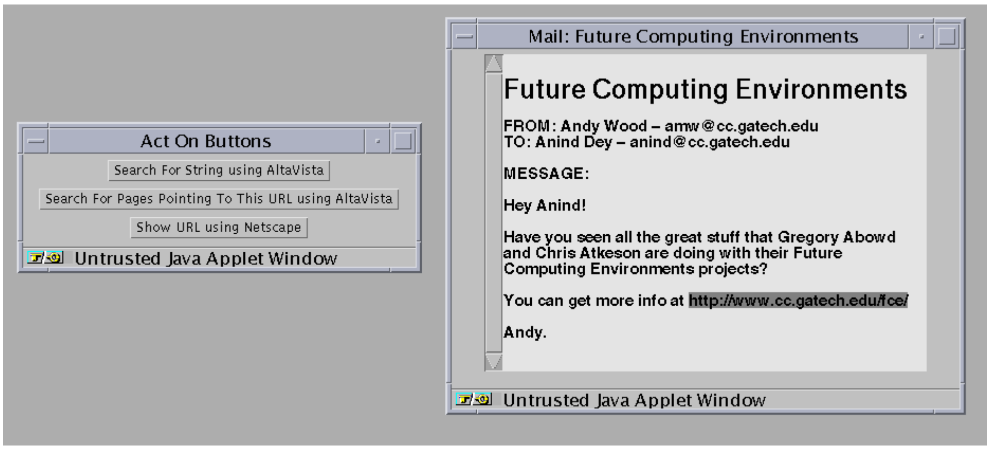



**Figure 4-6: User selects the URL and is offered suggestions by CyberDesk (Dey, Abowd, and Wood, 1998).**

By way of further example consider a situation where the highlighted text was a person’s name. In such a case CyberDesk could have offered to lookup the name using the locally installed Personal Information Manager.

Navigation in CyberDesk is document-oriented since the user starts with information (a document) and finds functionality (a service) that can act upon that information. This is distinctly different from searching a service and finding the proper information that the service can use. Whether the latter method of navigation is, or will be, supported is not discussed in the literature (Dey, Abowd, and Wood, 1998). The difference with CoolTown is that with CyberDesk the user stays at one place and navigates a virtual environment using a personal computer[120].

[120] The assumption is that the user uses a Web browser, which at the time of writing meant using a personal computer. The use of other appliances is not discussed in (Dey, Abowd, and Wood, 1998).

### 4.5.5. sView

SView is a project of the Swedish Institute of Computer Science based upon the vision stated explicitly in the introductory chapter of this thesis, namely that “the development of the World Wide Web […] has moved from publication of information to providing interactive services” (Bylund, 2001). With sView the research group tries to provide to an increasingly broad group of users continuous interaction with multiple services using various devices. To support user mobility this should be possible from different locations and situations. A further requirement is that users maintain control over their personal information.

The approach taken by the sView project is that instead of having a uniform client for accessing all servers (e.g. a Web browser), sView supports non-uniform clients by channelling all access to servers through a so-called ‘service briefcase’. Service briefcases contain “persistent and mobile images of the personal service environments of individual users. The service briefcase is what is actually saved when the user suspends the execution of a service environment to move or shut it down” (Bylund, 2001). When the user switches between devices it is the briefcase that is sent between network hosts[121]. By moving with the user in this way the briefcase is always nearby[122], and can be accessed from clients with different interface capabilities. Example interfaces are a Web interface and a Java Swing interface (Bylund and Waern, 2001).

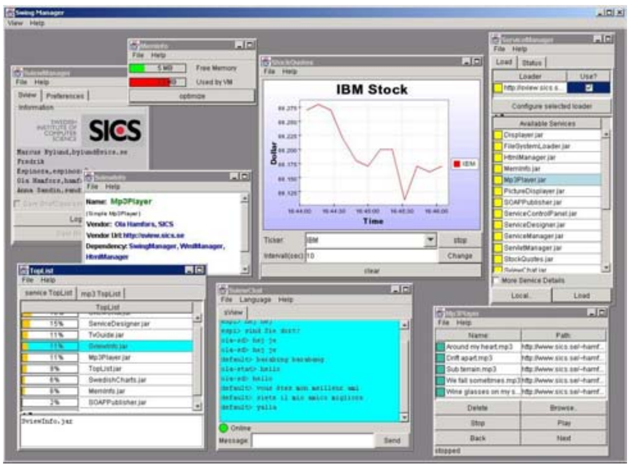



**Figure 4-7: Screenshot of the Java Swing interface of sView[123].**

[121] Called ‘service briefcase servers’.

[122] Where nearby means somewhere in the local network.

[123] From the sView Web site: http://sview.sics.se

The sView platform shows that there is yet another way to navigate services. If users have the sView software installed upon the device they are using, they are able to access their personal services briefcase stored somewhere on a nearby server. Thus they have direct access to their personal services without having to navigate through the Services Sphere to find their services. Within the briefcase the services are simply displayed as a large list (see screenshot above). They are not ordered in any particular way.

## 4.6. Discussion

The overview of the state of the art in navigating services shows that there is a growing awareness both in academia and industry of the difference between Information Space and the Services Sphere. The list of projects aimed at solving the problem of giving users access to network distributed services is growing steadily. The question lies in how far these projects explicitly address the issues of navigating services. It transpires that none of them makes that question their research. The ASP and Web Services approaches barely address the issue; they are mainly interested in the technological aspects of making services available to the users, not in providing them with a way to navigate a collection of services. Only a few *ad hoc* solutions, such as the IBM Web Service Browser, have been provided thus far. CyberDesk and sView take the approach that services travel with the users so they automatically have the right services at hand. In a CyberDesk environment the services move to, or are already available in, the virtual environment in which the user is, while with sView the services move to, or are already available in, the user's physical environment (they move somewhere nearby on the local network). While this approach solves the problem of finding services that are part of a small personal collection of services, it doesn't address how users can navigate, for instance, large collections of services, whether they are part of their personal collections or not. While services move around in the CyberDesk and sView environments, in the CoolTown environment most of the services are in some way fixed in the physical reality of the user. This is due to the fact that the CoolTown project is mainly focused on providing a bridge between the physical and virtual worlds. CoolTown doesn't address how users could access services that have no direct connection with physical objects (like a text editing service).

While the approaches taken by sView, CoolTown and CyberDesk are both valid and useful there are other ways to give users the opportunity to navigate services that have not yet been researched. If one assumes a user who navigates through a Services Sphere (the user is not moving physically), a useful research approach could be to explore the opportunities offered by combining different groupings of services organisations, metaphors, and the modes used (see the hierarchy of navigation in section 3.3).

All of the platforms discussed in this chapter build upon proven and successful technologies. ASP is chiefly a way of giving remote users a means of accessing applications in the desktop environment sense of the word. It assumes that the user sits behind a desktop computer that has the required software. Web Services go a step further and distribute not only the application but also the users over the network. The communication between the separate components is mediated by Web technologies such as the HTTP[124] protocol and the XML[125] data format. Similarly the sView platform is also mainly based on Web standards. The question is whether these platforms are appropriate for implementing the scenarios and explore the opportunities offered by combining different groupings of services organisations, metaphors, and the modes. As is often the case with using old technology to implement a new idea there could well be a mismatch between the desired goal and the means of implementing it[126]. When exploring ways to navigate services one should preferably not be limited by technology. Whether this can be prevented will be discussed when the technologies are chosen to be used for the research. But first the research for exploring the uncharted area has to be set up.

## 4.7. Summary

This chapter built upon the discussion in the previous chapter on navigating Information Space by focussing instead upon the questions raised by navigating services. A more detailed definition of what a service actually consists of than that one given in the introduction chapter was provided. This was followed by a discussion about what a collection of services (Services Sphere) is and how it differs from an Information Space. Next it was necessary to explain what the prerequisites are for a Services Sphere to become a reality. That the Services Sphere, and its accompanying navigational questions, is already becoming a reality was argued in the section discussing the state of the art in service-oriented environments. The overview showed that there are almost as many approaches to solving the navigational problems, as there are service-oriented environments. None of the approaches takes navigating services as their main focus point, leaving room for additional research.

Now that the research topics have been narrowed down to a few concrete open issues in the area of navigating services, it is time to construct hypotheses and choose a methodology for testing their validity. This will be the topic of the next chapter.

---

[124] Hypertext Transfer Protocol, a communication protocol to connect to Web servers.

[125] Extensible Mark-up Language, a standard for describing data.

[126] See '*The innovator's dilemma: when new technologies cause great firms to fail*' (Christensen, 1997) for some examples.

# 5. Research

*As the framework and some of its accompanying problems have been identified in the foregoing chapters, now is an appropriate time to pose a number of specific hypotheses the validity of which can contribute to the existing body of navigational theory. This chapter introduces the hypotheses and the methodology that will be used to test their validity.*

In 1994 Human-Computer Interaction expert Terry Winograd stated in an interview[127] that his field is driven by technology the potential of which in turn fired the imagination.

> *No particular piece of technology is an end in itself, but then somebody sees it and says 'here's something new we can do with this new technology that we couldn't do before'*
> *- Terry Winograd in (Preece, 1994)*

The motivation behind this research project has the same roots. When I was first confronted with the idea of a network of services my imagination was immediately sparked. I was excited by the prospect of (possibly) seeing solutions to problems afforded by such a network that are unsolvable using standalone desktop computer applications. My driving force has always been to make it easier for users to access, and use, all of the constructive functionality a desktop computer could offer — rather than to glorify a certain kind of technological innovation merely for the sake of innovation or progress. The new service-oriented technologies have created an opportunity to remove some existing barriers. This thesis is an attempt to further this process by seeing what barriers to progress exist, which was done in the foregoing chapters, and test whether they can be overcome using service-oriented networks, which can be found out by setting up a study. Such a study includes specifying the requirements for a useful and innovative system based upon the findings from the previous chapters, building such a system, and finally by evaluating it.

The goal of this research is an innovative, constructive, and informed investigation into the realm of services navigation, focusing upon a couple of specific issues. Among other things this means that the thesis does not contain the results of an extensive user study to learn whether the buttons in a particular interface are placed correctly. Rather the aim is to experiment with possible solutions to newly arisen issues and thus gain an insight into what will be the most fruitful direction for future research and development. The final result will therefore not, for example, be a detailed design guide for user interfaces but rather an overview of which are the most relevant issues and possible solutions in navigating distributed services. This thesis is

---

[127] As it appeared in (Preece, 1994)

intended as a guide for future research, which can include more rigorous and detailed user studies.

## 5.1. Hypotheses

The open questions of navigating services identified in the previous chapters are too broad to be manageable within the scope of this research. This section formulates three hypotheses that *can* be tested for validity within that scope.

The general research question as introduced in the introduction chapter is:

> *How does the transition from an environment existing of local documents and applications to an environment of distributed services and documents affect navigation?*

The theoretical overview of chapter 3 showed there is already a sizeable body of research that deals with the navigation of distributed documents. To extend this body of knowledge this thesis investigates the navigation of services. The hypotheses dealt with in the study are:

> **Hypothesis 1:**
>
> *There are differences between how users navigate in search of information (documents) vs. how they navigate in search of functionality (services).*
>
> **Hypothesis 2:**
>
> *A geographical metaphor is more useful for services with an 'actual geographical location' than it is for documents.*
>
> **Hypothesis 3:**
>
> *Different metadata is required from services than from documents to implement the different navigational user interfaces.*

Besides testing the validity of these three hypotheses the aim of this research is to provide specific details about the differences in how users search for information vs. functionality. Another aim is to provide a detailed overview of the required metadata for services and the differences from the metadata required for documents.

That the reader may know exactly what will be dealt with by this study and to prevent false expectations being raised, it is expedient at this point to make clear which issues will be passed over and what assumptions are being made. One assumption is that the user has a *fixed location* while browsing services. With the advance of mobile information devices it could have been that the user was roaming and therefore that the collection of services available to her was changing constantly. Such a scenario however would only serve to distract the study from its original focus and must be left to a later date. Once there is a good understanding of the specifics of navigating services from a static location, research into the effect of user mobility is both a logical and useful progression.

A key issue of which to be aware is that this research is not about navigation *within* a service, but rather deals with *inter-service navigation*. The fact that the services mentioned in this thesis are distributed across a network has an effect upon many interface design issues, again however that is not the focus or this research. Some of these questions have already been the subject of research for example by Hamfors, Neerincx & Pemberton, and Nylander & Bylund (Hamfors, 2001;Neerincx and Pemberton, 2002;Nylander and Bylund, 2002) while others remain open.

Some assumptions that may be less obvious, but which also affect the generality of the results, are about the user. The fictional user used during the study could well be called “Jane or John Average”, English speaking middle-aged westerners with no disabilities or other traits that affect the task of navigating services. This means the prototype does not account for any special considerations such as disabilities, age, sex, language, educational levels, culture, and so on. Such accessibility issues should definitely be taken into account when developing a navigational system for general use, but are less relevant for our current limited purposes.

## 5.2. Methodology

As mentioned above, the testing process should not only answer whether the hypotheses are true, but also provide specific details that support the answer. Such details include the differences in how users search for information vs. functionality, and an overview of the required metadata for services and the differences from the metadata required for documents. Finally the testing process should identify fruitful directions for future research.

The first part of the validation will be done by describing how a particular task is performed in a local setting (a setting without distributed services) and point out the differences in a distributed setting (including distributed services). Based upon the findings different prototypes of navigational interfaces will then be developed and tested. Varying with interfaces is necessary to find out whether, and how, the interface influences the validity of the three hypotheses. The description, building, and testing of the prototypes will serve to further validate the hypotheses and identify relevant details. The entire process consists of four distinct stages each of which has specific methodological requirements:

1. Describe how a particular task is performed in a local setting.

2. Describe different requirements for the same task in a distributed setting.
3. Prototype alternative navigational interfaces.
4. Tests with prototype to test the validity of the hypotheses.

It is impossible to anticipate all of the issues that may be encountered prior to building an actual prototype. This is why the development of the prototype, which is covered by the first three stages, will result in some insights as to the hypotheses. The development process is of itself an exploratory aspect of the research. Thus part of the results will come from, for example, the requirements analysis and the engineering task, while other results will follow from the evaluations.

## Stage one

A constructive method for stage one is to develop scenarios. Benner et al (Benner and others, 1993, 117-134) make it clear how scenarios can be helpful in our situation:

> *Scenarios are partial descriptions of system and environment behavior arising in restricted situations. They are instrumental to the following activities: describing and clarifying the relevant properties of the application domain, uncovering system requirements, evaluating design alternatives, and validating designs.*
> *- Benner et al (Benner and others, 1993, 117-134)*

Scenarios are valid not only for stage one according to this description, but also for the other stages. Stage two involves trying to envision future situations —again a situation for which scenarios are an excellent technique. Stage three uses scenarios to develop requirements for the prototypes, while the fourth stage will be an evaluation based upon scenarios. Scenario based research will be discussed in more detail later in this chapter since it's the main approach of this research.

## Stage two

Stage two highlights the differences in performing the same particular task in a distributed environment as opposed to a local environment. This entails looking into the future in an attempt to discern what the type of user, the environment, and the task will be. To highlight the most important differences as they relate to the issue at hand a present and a future version of a scenario will be developed.

Stages one and two have already been dealt with in chapter 2. It will be recalled that they discuss the differences between navigation in a local setting and navigation in a distributed setting. What has not yet been done is to specify the requirements for the prototype used in the scenarios. That task will be accomplished in the next chapter.

## Stage three

Stage three, which involves constructing prototypes of alternative navigational *user interfaces*, is where the requirements defined in stage two will be translated into concrete interfaces. This also requires a *platform* for navigating distributed documents and services that enables experimentation with alternative interfaces. The next chapter will construct a requirement specification for the platform and see whether existing platforms as discussed in section 4.5 are sufficiently flexible to allow this. The *platform* and the interfaces built on top of it will allow experimentation with different combinations of the layers in the hierarchy of navigation, as described in section 3.3, but this time for services. It should be repeated that the platform will only be used to implement the solutions in a distributed context, not in a local one. The precise requirements for such a platform are also discernible using scenarios.

The obvious tool to use is prototyping but there are many prototyping *techniques*. The important question to ask when designing prototypes is whether one wants to focus upon the range of features, and only implement part of their functionality, or whether to focus upon the functionality and build only a limited number of fully functional features. The former is referred to as a *horizontal prototype*, and could for instance consist of a "dummy" GUI (Graphical User Interface) that has no actions associated with its components. While the second option is referred to as a *vertical prototype* and could for example be a GUI having somewhat fewer components than the final version, but with such components having their full functionality.

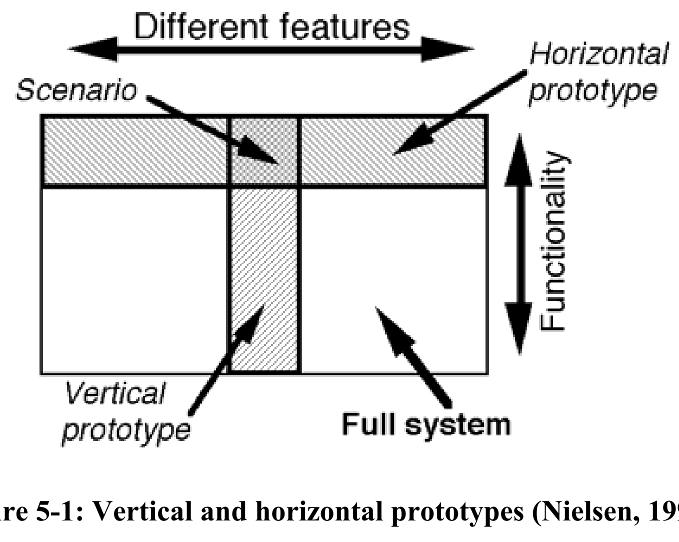



**Figure 5-1: Vertical and horizontal prototypes (Nielsen, 1994)**

Each prototyping technique has its own set of tools (Preece, 1994). A fitting method for testing the validity of the hypotheses is to build a vertical prototype using *evolutionary prototyping*. Evolutionary prototyping is an extensive form of prototyping that is a compromise between production and prototyping (Preece, 1994). It is a replacement for the waterfall model[128] of development. The process is incremental; rather than designing, implementing, and testing the system in one go[129], the developer designs an incomplete but highly flexible system and then iterates implementation, testing, and re-design. Each pass through the cycle should entail small changes as new features are added, until all requirements are met. Such a program easily adapts to changing requirements (Dodani, Hughes, and Moshell, 2002, 255-262;Zamperoni, Gerritsen, and Bril, 1995).

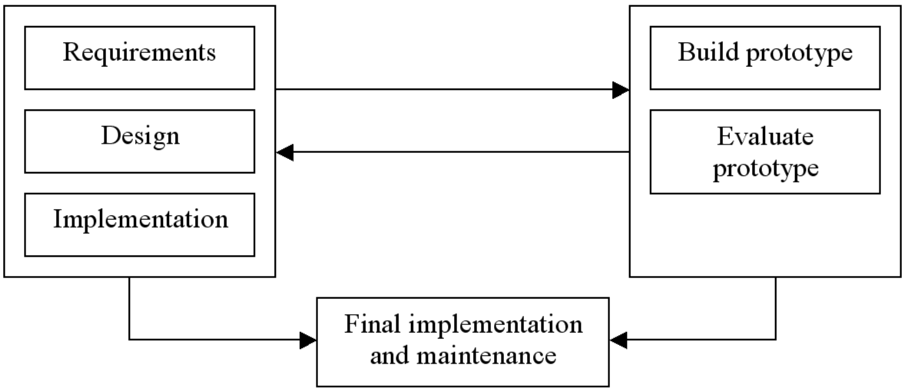



**Figure 5-2: Evolutionary prototyping development model.**

Because evolutionary prototyping uses only "production ready" services that will eventually be used in the final system, it has the advantage of ensuring that unexpected results due to the use of mock-ups do not occur. Its disadvantage lies in the fact that it's a less flexible method that encourages designers to focus upon one particular solution rather than exploring possible alternative solutions. As full functionality is not required for testing the hypotheses using a vertical prototype[130] is appropriate. This however is subject to the proviso that that part of the system that *is* implemented should be fully functional, thereby giving the user a complete experience of using the system.

---

[128] A software engineering term for a project that proceeds sequentially. Once each stage is completed, it cannot be reversed.

[129] As is the case with the waterfall model.

[130] For a difference between vertical and horizontal prototypes see Figure 5-1: Vertical and horizontal prototypes

## Stage four

Stage four involves evaluating the prototypes for feedback on the hypotheses. The evaluation approach will be a subjective and qualitative laboratory study since the evaluation is based on opinions and anecdotes collected in a controlled environment.

Evaluations should be driven by clear goals and appropriate questions (Basili, 1992) so to guide the evaluation of the prototype we will use the DECIDE framework specified by Preece, Rogers, et al in 'Human-computer interaction' (Preece, Rogers, and Sharp, 2002):

- **D**etermine the overall goals that the evaluation addresses.
- **E**xplore the specific questions to be answered.
- **C**hoose the evaluation paradigm and techniques to answer the questions.
- **I**dentify the practical issues that must be addressed, such as selecting participants.
- **D**ecide how to deal with the ethical issues.
- **E**valuate, interpret and present the data.

*Goal:*

The primary goal of the evaluation is to identify those issues that experts and users find important in navigating services. Other goals are to find out whether users think the metaphors used for navigation are helpful, and to identify metadata requirements.

*Questions:*

The more specific questions to be answered by the evaluation of the prototype are:

- Can the users find what they are looking for?
- Do the users have a clear understanding of the system they are using?
- Does the user feel in control of the system?
- Is the system easy to learn?
- Does the users get lost using the prototype to navigate services?
- Does the metaphor help users learning to navigate the system?
- What metadata is required from the services?

*Paradigm and technique:*

The evaluation paradigm will be first a predictive evaluation. In a predictive evaluation experts apply their knowledge of typical users, often guided by a set of high-level heuristics, to predict usability problems with a prototype (Preece, Rogers, and Sharp, 2002). It provides estimates by the experts about how well the prototype will perform for various kinds of tasks[131]. The techniques used will be interviews; first a small and unstructured interview followed by a lengthier and more structured interview. During the interviews the experts will be allowed to experiment with the prototype. Following Nielsen approximately five[132] experts will be interviewed freely both to review the prototype and to give their predictions on its effectiveness.

Using the experts' findings some modifications to the prototype will be made. A usability test of the modified prototype will then be conducted with about five users, an appropriate number given the qualitative nature of this thesis (Nielsen, 1992b). The users will be asked to perform some of the tasks taken from the scenario as developed in chapter 2, and a questionnaire accompanied by a set of open questions will be used to evaluate the user findings.

As well as a questionnaire and open questions the participants will be observed during their use of the prototype. This observation is intended to capture those aspects of the prototype's usage that might not be discoverable from the questionnaires and interviews. During the test sessions the participants will be asked to think aloud[133], explaining what they are thinking, what they are doing and why they are doing it.

*Practical issues:*

Practical issues are that the user base has to be as diverse as possible and that the usability experts have to be knowledgeable about navigating Information Spaces. The users should be handpicked to ensure a wide variation and a better representation of possible future users, but since the number of users to be tested will be limited it will be not be possible to get a good representative sample of possible future users of the prototype. This is not the problem it may appear to be as this is a qualitative rather than a quantitative study, and the results are not meant for generalisation. The experts used in the study should ideally come from diverse backgrounds. Experts with a background in navigation in general as well those with expertise on navigating Information Spaces are likely to have complementary views on the prototype.

---

[131] See (Preece, Rogers, and Sharp, 2002) for various predictive models.

[132] For a heuristic evaluation Nielsen recommends to "…use three to five evaluators since one does not gain that much additional information by using larger numbers". Source:

http://www.useit.com/papers/heuristic/heuristic_evaluation.html

[133] Thinking aloud is a useful technique for user testing if a sophisticated test-lab is not at your disposal. Nielsen (Nielsen, 1992a) showed that it is an effective technique for finding many important usability problems.

*Ethical issues:*

In order that both experts and users are comfortable with contributing to this research it will conform to the *Code of Ethics and Professional Conduct* of the ACM[134].

*Evaluate, interpret and present data:*

The interpretation of the collected data is an important step in the evaluation process. The reliability of the experiment depends much upon the control you have over it, however given the qualitative approach adopted it may well be difficult to repeat the experiment and achieve identical results. This does not mean that the results are useless; rather it highlights its qualitative nature and the attendant reliability limitations. Thus the results are limited and can only be generalised with great caution.

The result of each experiment will be closely examined for insights into the hypotheses, and when a particular issue seems to return to, or to touch upon, an important matter it will be discussed in light of the theory in the final chapters of this thesis. The most important findings will be summarised in the conclusion.

### 5.2.1. Scenario-Based research

Since scenario-based research plays such a prominent role in this thesis it is necessary to explain its characteristics in more detail.

Trying to discern how to construct future products that solve current problems inevitably bears the risk of blurred vision that results in vague statements and overbroad product descriptions. One approach to trying to solve this problem is the scenario-based research methodology described both by Wack (Wack, 1985) and by Carroll (Carroll, 1995;Carroll and Rosson, 1991)[135]. In scenario-based research it is assumed that the future cannot ultimately be known with complete certainty. Taking this as their starting point scenario planners set out to think about various potential

---

[134] ACM Code of Ethics and Professional Conduct: http://www.acm.org/constitution/code.html

[135] Erskine et al (Erskine, Carter-Tod, and Burton, 1997) give a clear summary of the ideas discussed by Carroll and Rosson which I will repeat here:

"Carroll and Rosson treat design as inquiry. Their methods are meant to make explicit those assumptions and activities of the design process which are usually implicit. The primary argument is that designers use scenarios (walkthroughs of a design artifact in use) to informally test the usefulness and usability of design artifacts. From the test, the designer draws conclusions about the artifact and modifies the design. In scenario-based design, the first step is that a scenario is written down as a detailed narrative. Next, claims are made about the usability and usefulness of particular artifacts envisioned in the scenario. These claims are also recorded in a manner that maintains their link to the scenarios they analyze. This process of scenario construction and claims analysis is conducted as an iterative cycle. In the end, the accumulated scenarios and claims constitute the design's description and rationale."

futures that could emerge. The process consists of a range of techniques (research, brainstorming, story telling) and imagined stories delineating the boundaries of what *could* occur in the future (Laubacher and Malone, 1997). Bruce Togazzini (Tognazzini, 1992) pointed out how the scenario-based technology can give the researcher a way to define and develop a sense of user space. It forces systems designers to consider both a range of users and a range of activities for which the system could be used. "User space" in this case being defined as the variety of users, work, and environments in which the interaction with the system can take place. Scenarios are then used to make concrete combinations of these three dimensions[136].

Carroll gives an overview of how the scenario-perspective compares to previous views (Carroll, 1995) and which shows what a good scenario should offer. That such a methodology is appropriate to this research is emphasised by the existence of "envisioned outcomes."

| The scenario perspective | Previous views |
|---|---|
| Concrete descriptions | Abstract descriptions |
| Focus on particular instances | Focus on generic types |
| Work driven | Technology driven |
| Open-ended, fragmentary | Complete, exhaustive |
| Informal, rough, colloquial | Formal, rigorous |
| Envisioned outcomes | Specified outcomes |

**Table 3: Scenarios compared with other ways of system description (Carroll, 1995)**

Thus rather than designing software by taking as a starting point a list of requirements designers start from the activities that need to be supported. They then allow descriptions of those activities to drive the rest of the process. Based upon these properties scenarios offer a couple of advantages that make them a good choice for this research. Scenarios have the following desiderata[137]:

- **Clarify assumptions.**
  By agreeing on usage details, development team members formalise their assumptions so that there are fewer unpleasant surprises later.
- **Fully explore the design.**
  Scenarios provide a vehicle for exploring all of the important aspects of the interface.

[136] See Preece (Preece, 1994) for a good example of a scenario.

[137] Quoted from http://www.user.com/scenario.htm

- **Improve understanding of the user.**
  Research into types of users and their needs is needed to complete scenarios. Knowing who the users are and what they understand about the task helps us design a product they can use.
- **Provide a context for reviewers.**
  Detailed scenarios provide a context for understanding the features of interface specs, functional specs and other project documents.
- **Provide direction for other activities.**
  Scenarios represent the important tasks that major classes of users are expected to perform. They are useful in usability testing, documentation, QA and others areas.

Scenario-based system design is a good method for developing multiple prototypes and testing them on a limited scale within a short time span (Nielsen, 1994;Nielsen, 1993). The reason it is used for this research is that the resulting structured thinking about scenarios clearly indicates what the important aspects of the system will be. Furthermore it offers a rapid development process that is constantly monitored by user feedback. Together these two advantages minimise the chance of arriving at the wrong destination without the possibility of return. Moreover the constant user feedback filters out many potential errors early in the process.

As explained in section 5.2 scenarios play a role in each of the four stages in the research process. More concretely, the scenarios are used to:

- *Explain the new situation.*
  First, based upon brainstorming sessions and background literature studies a few scenarios were developed to explain the envisioned changes[138].
- *Highlight the research topic of this thesis.*
  Next scenarios were used to highlight new research topics.
- *Acquire prototype requirements.*
  Subsequently scenarios will be used to acquire the requirements a prototype has to fulfil.
- *Implement scenarios.*
  Also when implementing a use-case, scenarios will be used.
- *Testing.*
  Finally for testing the users will be asked to perform some of the tasks in the scenarios.

---

[138] See chapter 2.

The first two applications of scenarios have already been dealt with in the thesis thus far; the remaining three will be the topic of the following chapters.

## 5.3. Summary

This chapter summarised the findings of its predecessors by formulating a general research question. This general question was in turn split into three concrete hypotheses that can be tested for validity within the scope of this thesis by conducting a small study. Next the methodology for answering those questions was chosen. It transpired that scenario-based research was the best means of finding relevant properties of the application domain, uncovering system requirements, evaluating design alternatives, and validating designs. Evolutionary prototyping was chosen for constructing a horizontal prototype that fits the scenario. For evaluating the implementation of a scenario using the prototype a predictive evaluation combined with user tests will be conducted.

The next chapter explains what kind of platform is needed to implement the different prototypes and scenarios. The requirements were identified with the analysis of the two scenarios in chapter 2. After the platform has been described the implementations will be discussed in detail in the chapter that follows. The last few chapters are concerned with the evaluations and conclusions.

# 6. Services Platform

*To be able to construct different interfaces[139] for navigating services the underlying platform, called Services Platform, must first be built. The requirements for such a platform will be obtained from the scenarios.*

## 6.1. Requirements

The requirements for the Services Platform will be extracted from the analyses of the scenarios in chapter 2. However there are many decisions to be made that, while they will not directly influence the implementation of the scenarios, are relevant to improving service navigation usability. An example of such a decision is whether to use a technology that enables a service to be "fault tolerant," i.e. whether it is capable of restoring itself after a failure. The presence or absence of fault tolerance may not appear to be of direct relevance to the implementation of the scenarios. However it can have considerable relevance to ensuring that a system is as user-friendly as possible using existing technologies. Such decisions are in any case part of the design process and it would be imprudent to consider only those requirements needed to fulfil the scenarios. I therefore first discuss those requirements that flow directly from the scenarios and more general requirements thereafter.

### 6.1.1. Scenario requirements

First and foremost the Services Platform should enable an implementation of the future versions of the scenarios given in chapter 2. The focus of the requirement analysis will be mainly upon the 'Travelling Pictures' scenario as that scenario was developed with the issues of navigation in mind. The Services Platform should enable both the services and a way of navigating them to be developed[140].

In the first scenario, 'The Trip,' a wide range of devices was used to access an even wider range of services. To give but two examples desktop computers were used to access personal agents, and mobile phones were used for voice messaging. As the scenario's aim was to illustrate the possible effects of the changes introduced in the first chapter, it did not specifically address navigational issues. However it did address the fact that the nature of the navigational task changed as both the devices used for navigation, and the entities being navigated changed.

---

[139] Remind that the term 'interface' is used for interfaces for humans, unless mentioned otherwise.

[140] It should also be borne in mind that the scenarios specify the devices that provide services themselves or are used for accessing services. See the analyses of both scenarios in chapter 2 for an overview of all the services and devices.

The second scenario, 'Travelling Pictures', was focused more upon the navigational task in a future situation where particular circumstances now exist. The scenario dealt with four tasks:

1. Give a slideshow.
2. Retouching pictures.
3. Print pictures.
4. Update the travel log.

Each task touches upon different navigational problems, for the scenario to become a reality each task must be accomplishable.

## Give a slideshow

The first task is to give a slideshow using a digital camera and a variety of displays. For this to be doable two devices such as the digital camera and the TV need to work together. The underlying assumption here is that several networking technologies, which will be discussed later on in this chapter, are in place.

In the scenario when the need arises to connect devices the user is able to use both the display on the camera to find the TV, or to launch an external browser and find both the camera service and the displaying service and make them work together. Again, the Services Platform should enable users to experiment with both approaches.

## Retouching pictures

The difference between retouching pictures and the previous task is that it requires a local device to work together with a *software service* that resides somewhere on the Internet. The Services Platform should enable various means of connecting the two. In the scenario the actor had a list of favourite services that she could combine with local services. However it could have been that she would have had to look for the service on the Internet.

Another issue this task points out is that there is a difference between starting with a service (image editing) and finding some data upon which it can act and starting with data, in this case images, and finding a service that can act upon it. The Services Platform should support both alternatives.

## Print

The task of *printing the pictures* is essentially the same task as displaying them. In the scenario however, other than a facility to put it in standby mode the printer has few interaction capabilities, it has neither a display, nor a speech interface to enable navigation. Given such limitations it is impossible for the user to start with the printer and use it to locate printable material. The Services Platform should enable such devices to be part of the network and provide a way for users to access the services of those devices. Another device with service-browsing capabilities would be needed to use devices without interaction capabilities. This is a requirement for the scenario to become reality.

**Update travel log**

Updating the travel log touches upon the issues of storing and retrieving the data used by a service. In the scenario the tasks of storing the entries for a travel log, and of publishing them, are both automated. The logging service automatically stores and retrieves the log from the Internet when needed. Such automation is one solution; another would be to allow the user to choose the storage location. Both would require a way of organising and navigating data. The Services Platform should enable both automated and manual operations.

Finally the Services Platform should not only enable the implementation of the scenarios, but also be sufficiently flexible to allow experimentation with various interfaces[141]. Such flexibility would allow exploration of various solutions to the research questions, and could be provided by selecting appropriate technologies. These technologies will be discussed below following an overview of the remaining requirements.

### 6.1.2. Handle networking issues

Developers of services that use network communications need to be aware that the network introduces a set of extra issues that are not present when developing standalone applications. These are not only technical issues such as latency[142] and fault tolerance, but also non-technical issues such as a user-friendly handling of network failures. Unfortunately many of these aspects are often neglected or simply not realised, resulting in products that are sub optimal both technically and in terms of their usability. These aspects are:

- **Latency.**
  The delay in communication over the network.
- **Partial failure.**
  Some of the resources the system depends on might be down or unreachable.
- **Node failure.**
  A resource upon which the system depends on might fail. This is sometimes also referred to as a single point of failure.

---

[141] Meaning that it should enable different combinations of the ‘organisation’, ‘metaphor’ and ‘mode’ layer of the hierarchy of navigation (see section 3.3).

[142] The time between initiating a request for data and the beginning of the actual data transfer.

- **Network partitioning.**
  Some parts of the network might be unreachable to other parts.

Common misconceptions about these issues are clearly summarised in Deutsch's eight fallacies of distributed computing (Deutsch, 2001). These networking issues and their accompanying fallacies are important for anybody building distributed system and interested in the usability aspects of it.

1. **The network is reliable.**
   This fallacy assumes there will always be a network and that it will always function correctly. The effect for usability engineers is that systems will have to deal with this unreliability in as user-friendly a way as possible. For instance the user should be made aware that some of the actions require communication across the network, and that this could fail. In the event of such a failure it should be handled in a way that minimises interruption of the user's interaction.
2. **Latency is zero.**
   Networks by definition introduce latency, which may vary depending upon the topology of the network. Response time has long been a topic in Human-Computer Interaction research (Dix, 1987, 215-228;Dix, 1999;Miller, 1968) because to a great extent user satisfaction with a system is dependent upon the responsiveness (Cooper, 1995;Mayhew, 1992). The latency introduced by the network will affect every action involving network communication, consequently the chance of delays occurring that a user might notice increases for networked applications. Clearly therefore network induced latency is a seemingly purely "technical" issue that has to be dealt with at a usability level also.
3. **Bandwidth is infinite.**
   This fallacy has been the starting point for many discussions about (often unrealistic) scenarios involving networked services (Gilder, 2000). The fallacy already resulted in user-unfriendly software[143]. Designers of distributed systems will need to keep in mind that the amount of data that can be send over the network is limited. Thus instead of full screen streaming[144] videos the designer might have to "make do" with images that are updated every few seconds. Another example of such design compromises imposed by

[143] Anyone who has ever watched an online video will readily admit that bandwidth is a bottleneck for a unspoiled user experience.

[144] The client buffers a few seconds of video before it starts showing it on screen, which compensates for momentary delays in data delivery.

bandwidth availability might be that a designer electing to use midi music[145] rather than recorded music.

4. **The network is secure.**
   Security is an issue for all computer systems and networking adds an extra source of insecurity, as many of them are routed through public areas. This has become an even greater problem with the surge[146] in use of wireless networks (Ramasami, 2002) a factor which goes some way towards explaining the rise in security expenditures[147]. Cleassens et al (Claessens, Preneel, and Vandewalle, 2002) have listed the current security issues affecting the Internet. They point out that while security involves more than technology alone, there are definitely security issues that reside purely at the technical level.

   For usability engineers the consequence of the new security issues is that they need to make users aware of these issues, but in such a way as not to severely affect user confidence in the service. However if existing Web browsers are a reliable indicator this is often not done. Many current Web browsers “hide” the settings that affect security issues somewhere deep in their ‘Settings Menu’ leaving the user unaware of possible security threats implicit in leaving system defaults unaltered. Furthermore many users of existing networked applications are ignorant of whether they’re communicating over a network or not and of whether or not their data is secure (Claessens, Preneel, and Vandewalle, 2002). Unfortunately security matters are far from easy for the average user to understand. In response to the threats and to user indifference to security matters software developers often choose a common denominator for security issues so that users don’t have to concern themselves with these low level details. Such a response shows that security matters and usability aspects directly influence each other.

5. **Topology doesn’t change.**
   Given that existing networks consist of *loosely* coupled entities one has to assume that the topology of the network will change. Computers, printers and

---

[145] Musical Instrument Digital Interface: A standard protocol for the interchange of musical information between musical instruments, synthesizers and computers. It defines the codes for a musical event, which includes the start of a note, its pitch, length, volume and musical attributes, such as vibrato. Since midi files only describe the music they are very small compared to recorded music. The difference is of course that it’s a replay of the song with the midi instruments, not the original.

[146] A good online reference for up to date statistics on all things Internet is the Cyberatlas: http://cyberatlas.Internet.com/big_picture/stats_toolbox/article

[147] Source: Dataquest:

http://cyberatlas.Internet.com/big_picture/applications/article/0,,1301_969961,00.html other devices are added to the network in a continuing fashion constantly changing its topology. The latest network technology advances are often focused upon a constantly changing topology. Networking technologies such as Jini (Waldo, 1999, 76-82), peer-to-peer networks (Oram, 2001), Universal Plug and Play (Chen and others, 2001) and BlueTooth (Haartsen and others, 1998) all involve dynamic networking. The increasing use of such technologies serves to illustrate the fallaciousness of the assumption that the network is static and unchanging.

Topological changes also affect the usability aspects of services (which are distributed across the network). Parts of the network might disappear during a session meaning the user can no longer access some functionality. For example; a device such as a printer could be removed from the network without notification. A rise in the use of services implies that topological changes will occur far more frequently than when using a stand-alone desktop computer. To ensure a seamless user experience users need to be made aware that this can happen, so it won't confuse them *if* it happens. How the users are made aware depends on the situation, it can for instance be the service or the service browser that makes them aware.

6. **There is exactly one administrator.**
   A single person does not administer large networks. The Internet is a good example of why this is a fallacy. It consists of a very large number of sub networks, each of which is administered by different people. Peer-to-peer networks take this to an extreme as a different person could administer each network node.

   A consequence of such diversity is that different parts of the network might have different access policies. One part might require authentication, while another might limit the amount of time users are permitted to access its resources. Yet another consequence could be that users might be less trustful of the administrators of an unknown network segment and might for example refuse cookies[148], or have set other security preferences defaults that refuse to give information needed to access particular services. Together the consequences affect the use of services that are distributed over such networks. Designers and consumers of such services need to take the aspects into consideration when developing and accessing them.

7. **Transport cost is zero.**
   Using a network costs money and a good business strategy for developing distributed systems needs to take account of this, which has an impact upon usability. To an extent this depends upon the service for some services don't

[148] Data created by a Web server that is stored on a user's computer.

require much bandwidth. Instant messaging[149] for example does not require a lot of bandwidth as it is (currently) only used to send small text messages between users. On the other hand video conferencing requires streaming video, something that does require a lot of bandwidth.

Interfaces with a lot of graphics and sound are bandwidth intensive. However in order to minimise transport costs some service providers could opt to implement smaller and simpler interfaces, sacrificing some usability in the interest of minimising costs.

8. **The network is homogeneous.**
Some networks, and the Internet in particular, consist of sub networks with different capabilities. Parts of the Internet for example run on fast public networks while other parts run on slow and unstable privately owned wireless networks[150]. The increasing share of wireless networks in the Internet (CyberAtlas, 2002) will ensure that difference in capabilities between the each sub network becomes more visible and more important.

   Differences between networks will need to be taken into account by service designers, and the choices they make will in turn affect the user experience. For example; to be accessible from as wide a range of network segments as possible (from mobile phone networks to high-speed local area networks) a service designer might choose to take a "lowest common denominator" interface as a starting point, just to keep the amount of data to be send to the users minimal. Thus users accessing the service *via* a high performance network segment might have to use an interface with fewer usability features than their network is in fact capable of handling.

All the issues mentioned in this section need to be taken into account when building systems that include a networking component.

### 6.1.3. Platform independence.

The 'Trip' scenario showed that the possible number of devices used for accessing services may be quite large. The more devices upon which the Services Platform runs the more choices are available to users. To accomplish this the Services Platform

---

[149] A conference, over the Internet, between two or more people. Instant messaging is not a dial-up system like the telephone; it requires that both parties be online at the same time. You have to put the names of people you want to instant message with in a list, and when any of those individuals log on, you are "instantly" notified so that you can begin an interactive chat session. AOL's Instant Messenger (AIM), Microsoft Messenger, ICQ and Yahoo! Messenger are the major instant messaging services.

[150] Wireless networks typically have a lower bandwidth than wired networks (Tanenbaum, 1996).

should function as independently of the underlying operating system as possible. Thus it is desirable that the Services Platform should be capable of running under operating systems as diverse as that for mobile phones to Unix, and from Windows, or some variation thereof, to that for palmtop devices.

Given that software platforms are often associated with a specific hardware platform[151], detaching the Services Platform from the underlying operating system would also help the user to have freedom of choice of hardware platform. Such freedom of choice is desirable, as it would ensure that the user could use the Services Platform from many devices. This ensures a larger the number of potential test environments.

Hardware platform independence also ensures that the developer doesn't have to develop completely separate versions of the service, thereby decreasing the probability of introducing errors. Separate interfaces for the different platforms, whereby the underlying system doesn't have to change, should be sufficient.

### 6.1.4. Minimal administration

Taking the burden of system administration from users means that they would no longer have to concern themselves with such issues as the installation and maintenance of services, device management, or backing up their data. The introductory chapter showed that that this would save users both effort and frustration. Ideally users should no longer have to concern themselves with upgrading applications or with where to store their data (if they prefer that). Also, a user should have the choice to perform a 'save' secure in the knowledge that their data is in 'safe hands[152]' and will be immediately available the next time it is needed. The need for this was highlighted by the future version of the "Travelling Pictures" scenario when the actor wanted to edit an image. She navigated to her virtual office to launch an editor and edited the image. No installation or maintenance was required for this. A specialised company provided the photo editing service, part of whose function is to ensure that its clients always have the latest version of the service available[153].

For a user-friendly situation network administration should no longer be a part of the daily routine of using computers. Attaching a new (network enabled) device to the network should no longer require the user to configure many parameters or to install device management software. It is a job that takes specialised training, something that

151 Examples are how the Mac OSX operating system only runs on Apple hardware and PalmOS only on a limited collection of PalmOS compatible hardware.

152 What 'safe hands' are depends on many factors, of which an overview is beyond the scope of the thesis. The point is that users should feel their data is safe.

153 As a further option they could have provided the service of storing the images for her as well.

most users didn't have. Thus users are not to blame for their difficulties with such tasks; the fault lies primarily in excessive system complexity. Furthermore given that one of the assumptions of this research is that the number of devices connected to the network will increase rapidly (see chapter 1) it is likely that this will become an even greater problem. A solution to this problem is for the network to be self-configuring. The 'Travelling Pictures' scenario gives an example of such a need. In the scenario the actor attached a digital camera to the network in order to give a slideshow. So that the user's task would be as simple as possible the configuration completed without user intervention.

### 6.1.5. Rich user interfaces

In most cases services require more user interaction than that involved in merely reading a document (Nielsen, 2000a), this results in higher interface requirements. Take for example, the difference between *editing* text or *reading* it. The interface requirements for the former are undoubtedly higher. The difference in interaction requirements not only concerns the look of the interface, but also aspects like accessibility[154] and multi-modality[155]. This variation in interface requirement means each service could well have a unique interface.

The Services Platform should be able to provide users with an interface that best fits their needs[156]. This means that it should be able to render widely different interfaces for different services. It also means that the Services Platform should be able to choose the appropriate interface for the device it runs on. Thus if the Services Platform were to be running on a PDA the interface for the service should be the one that is optimised for a PDA. This might mean picking a platform specific interface, or if such an interface is not available, picking the 'best' available alternative.

Therefore the Services Platform should offer the possibility of creating rich interfaces. Since this research is human-centred it is important that the Services Platform offers no hindrance to this. Among other desiderata the Services Platform should have a good library with all the components for building such interfaces. It should not only have all the 'right' buttons and labels, but also support accessibility, internationalisation and multithreading.

*Accessibility support* means that the Services Platform helps people who are physically or visually impaired. Accessibility support is not only relevant at the level

---

[154] Accessibility concerns helping the impaired with using software.

[155] Multi-modal means that the user can use different modes (text, speech, etc) to interact with the software.

[156] If the user is for instance colour-blind and has specified that in their preferences the Services Platform should pick the interface that is designed for colour-blind people — assuming that such an interface is available.

of the individual interface components, but also at the level of the whole interface. Interface components (such as a button) might have accessibility support built in, it might for instance have a sound icon[157] associated with it to make it accessible for the deaf. This makes that the developer doesn't have to worry about accessibility at the interface component level. For navigating the overall interface (consisting of multiple interface components) on the other hand developers will have to take care of accessibility issues themselves. They for instance have to decide which button goes where in the interface.

*Internationalisation* means that the interface adapts itself automatically to accommodate different nationalities. Thus such aspects as the language of the labels and help, the colours, the sounds might change in accordance with the nationality specified by the user. Automating the process of adapting the interface to different nationalities would save interface designers time while ensuring that the usability issues raised by international differences have been dealt with.

*Multithreading* allows multiple streams of execution (called 'threads') to take place concurrently within the same program, each thread processing a different transaction. At the interface level this means that for each action involving the interface a separate thread is created. For instance where a particular action requires network communication it is useful to handle the communication in a separate thread so that other threads can directly handle the rest of the actions involving the interface. This is important because it keeps the interface responsive, a factor of especial importance when a selected action takes a long time to complete.

[157] A word that when spoken triggers the component. An example is an 'OK' button with "OK" as the sound icon. This way blind users, or users unable to read the screen at the moment of use, can use speech recognition to push the button.

### 6.1.6. Accommodate user diversity

Users differ in many aspects and both services, and the Services Platform to navigate them, should accommodate such user diversity. Schneiderman points out (Schneiderman, 2000, 85-91) that this involves accommodating users with different skills, knowledge, age, gender, abilities, disabling conditions (mobility, sunlight, noise), literacy, culture, income, to name but a few (see drawing below).

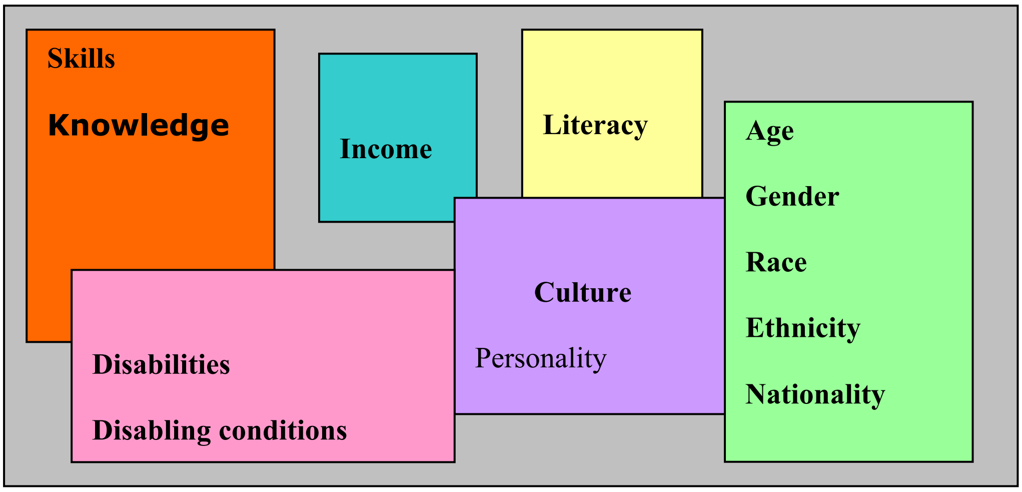



**Figure 6-1: User diversity.**

While Schneiderman's article mainly focuses on Web pages his points are equally valid for services. When one takes into account that in general services require more interaction from the user it is likely supporting user diversity will become even more important. The drawback to this is that accommodating users with widely different skills will require additional design effort from service providers given that the interaction patterns with services are more complex.

Although the need to accommodate user diversity focuses upon the services themselves rather than upon navigation *between* them it is relevant to the Services Platform as well. To the extent that the platform is used for navigating between the services such an accommodation is reliant upon the implementation of the service - but should not be limited by the platform. If developers were to decide to make special provision for deaf users who want to use the service while travelling, they (the developers) shouldn't be limited by the Services Platform used to access the service.

Designing for a wide diversity of users could have a negative effect upon usability if different versions of the service for different users are not clearly differentiated, as (Odlyzko, 1999) points out:

> *"…it appears necessary to recognise that flexibility and ease of use are in an unavoidable conflict, and that the optimal balance between those two factors differs among users. Therefore systems should be designed to have degrees of flexibility that can be customised for different people."*
> *- (Odlyzko, 1999)*

In other words more experienced users might want to use a more flexible and feature rich version of a service, while inexperienced users might want to start with a limited version that is easier to use.

### 6.1.7. Afford multiple navigation techniques

The requirement of affording multiple navigation techniques should be interpreted in the light of the hierarchy of navigation[158]. One such navigation techniques could for instance be semantic navigation, which could involve a 2D interface using a spatial metaphor. The Services Platform should be flexible enough to allow experimentation with different means of navigation. This is needed for not only research into alternative ways of navigation, but might ultimately be needed for different user preferences in navigation techniques.

### 6.1.8. Code mobility

The Services Platform should support code mobility, which is:

> *The capability to reconfigure dynamically, at run-time, the binding between the software and their physical location within a computer network.*
> *- (Carzaniga, Picco, and Vigna, 1997)*

This means code can be dynamically loaded from the network as it is needed. This requirement stems from the fact that moving computation closer to the user has clear usability advantages for the user[159] (Dix and others, 2000), as well as enhancing infrastructure efficiency (Halls, 1997). In 'Places to stay on the move' (Dix and others, 2000) Dix et al point out that despite the focus of research in code mobility on low-level communications, that there are stimulating results for usability research. One

---

[158] See section 3.3.

[159] An example could be the fact that the software is more responsive because mobile code makes local execution of tasks possible, making network communication unnecessary.

goal that can be accomplished using mobile code is the reduction of latencies associated with the 'feed through' of collaborative systems.[160]

Most of the networking issues mentioned in the previous section would play a role in any collaboration between users. A larger network could introduce more latency or a bigger chance of failure. This in turn affects the feedthrough time. In such situations mobile code proves useful. If, for example, two users could communicate directly with each other without going through a backend system the feedthrough time could well be significantly reduced. The diagram below illustrates this point:

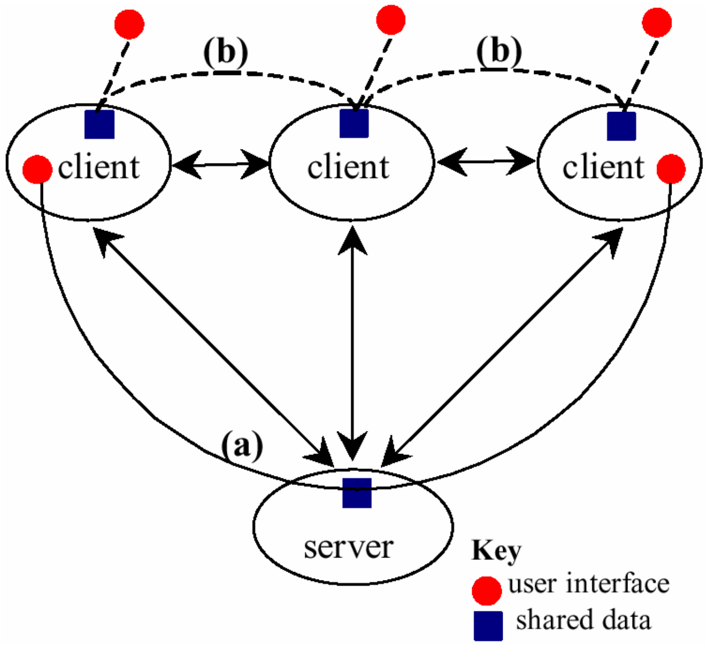



**Figure 6-2 Collaborative networked system (from (Dix and others, 2000)).**

In systems without mobile code all communications between users has to go through a server, this would introduce networking issues such as latency and a bigger chance of the network failing[161]. With mobile code the computation can be moved closer to the users giving them the opportunity to communicate directly with each other while still using the same system. Thus using mobile code could reduce or even completely remove some of difficulties inherent in networking. To give but two examples, feedthrough time would be reduced, and users autonomy would reduce the impact of a

---

[160] Feedthrough is the ability of users to see the results of one another's ongoing work. All aspects of the networking medium between them affect it.

[161] See the beginning of this chapter for more networking issues.

temporary server failure. Such advantages would improve usability (Dix and others, 2000).

A disadvantage of code mobility is that it opens up security risks. By letting code from the network run locally without intervention, users risk running malicious code. The security model of existing platforms supporting mobile code, for example Java, try to minimize the hazards by preventing the downloaded code from performing unprivileged actions[162].

### 6.1.9. Network-centrism

The third requirement is that the Services Platform should be network-centric since the focus here is on navigating services that are distributed across the Internet. Networking support should therefore be an inherent component of the platform. Were this not to be the case the Services Platform could be too limited in its capabilities and its subsequent further development to allow for networking support take too much time.

### 6.1.10. Requirements summary

To conclude this discussion of the requirements it is valid to ask whether the state of the art in service-architectures described in section 4.5 can be used for this research. If such were to be the case it would save the effort involved in developing the Services Platform. Regrettably none of the currently available technologies addresses all of the requirements. Thus in order not to escape technology-imposed limitations it is necessary to design a specific platform for services, the Services Platform. The advantage to this is that the design process in itself is likely to yield insights into the research questions.

## 6.2. Technologies

As the main requirements for the Services Platform have now been identified this section will explain why certain technologies were chosen for building it. While it is true that there are other technologies available that could have accomplished the task, a discussion of these alternatives is not the primary focus of this thesis. Moreover the industry and research field is so dynamic[163] that keeping up with the latest technological developments would have taken too much time. The technologies used were primarily chosen for their ability to deliver certain levels of functionality and

[162] For example deleting files without permission.

[163] For example during this research the industry went from inventing the term 'Web Services' (two years ago) to a situation where the majority of news items about the industry dealt in some way with Web Services.

meeting the specified requirements. Another selection criterion was the level of my own expertise with the technologies.

### 6.2.1. Java

The Java platform consists of a programming language and a virtual machine[164]. By using a virtual machine the Java platform is an abstraction of the underlying operating system (Arnold, Gosling, and Holmes, 2000). An application written in Java runs on top of this virtual machine, so that it runs on each operating system for which a virtual machine has been developed. Therefore Java is independent of the underlying operating system and hardware platform.

Java also fulfils the requirement of being able to move functionality across networks, as one of its development goals was to support code mobility. Most readers will be familiar with the Java applets found on many Web pages. Such applets are an example of the use of mobile code. The user visiting the Web page loads the code for the applet and executes it locally[165]. Any additional code that the user might need can be downloaded and executed as needed.

Java is also a good choice when it comes to network capabilities. It was built to make implementing common networking tasks as simple as possible for the developer (Arnold, Gosling, and Holmes, 2000). Network programming is made simpler for the developer not only because they can use the standard Java classes to take care of most of the difficult networking tasks, but also because important networking issues such as security, fault tolerance, and versioning were taken into consideration while Java was being developed. This prevents developers from having to take care of those issues themselves. Finally Java has excellent library for building interfaces, which includes many interface components that support accessibility, internationalisation and multithreading. All of the characteristics mentioned above make Java a good choice upon which to build the Services Platform.

---

[164] A virtual machine is a software program that emulates a hardware system. An application written to run on the virtual machine will run on any hardware system the virtual machine runs on.

[165] The technically inclined might wonder about the security issues involved here. Java uses a 'sandbox model' that executes the code in a protected environment to test whether the code can be trusted. If so, the code will be granted permission to execute on the client.

### 6.2.2. Jini

The Services Platform will be built using Jini networking technology, which is defined as (Arnold and others, 1999):

> *A set of Java application program interfaces (APIs) to enable transparent networking of devices and services, thus eliminating the need for system or network administration intervention by a user.*

Jini is built on top of Java, which in turn uses a virtual machine to abstract the developer's code from the underlying operating system (see previous section), thereby making it platform independent. The figure below illustrates the technology stack:

| |
|---|
| **Jini** |
| **Java** |
| **Operating system** |

**Figure 6-3: Jini in the technology stack.**

Jini will be used for this research because the vision behind Jini since its inception has been to eliminate the need for administration. One of Jini's goals is to enable services to cooperate spontaneously without the need for user intervention. This spontaneous inter-service cooperation takes place irrespective of whether these services are being offered by hardware (e.g. a Jini enabled television) or software (e.g. a text editing service). Thus a network of Jini services should automatically manage services appearing and disappearing from the network, and enable cooperation between the services. Such automatic service management has the following implications:

1. New services that are connected to the Jini network are automatically configured to work collaboratively with the rest of the network. This is achieved by having one or more lookup services running in each Jini network keeping track of all of its services.
2. It enables the network to manage the removal of services, and to do so gracefully[166].

---

[166] A service can disappear because the device was removed, the software crashed, someone stopped the service or because part of the network failed.

Jini addresses the problems of distributed computing[167] using a set of simple interfaces and protocols. The sequence of events when a service is being connected to the Jini network is as follows:

1. The service provider sends a *discovery packet*[168] that tries to find a *lookup service*[169] in the network.
2. If the packet finds a lookup service the latter will send a *registration object*[170] to the service provider that it can use to register its service with the lookup service.
3. The service registers itself on the lookup service using a *service item*[171] that represents its capabilities.
4. Upon registration the lookup service returns a lease to the service provider that it must renew at set periods, if it fails to renew the lease its service item will be removed from the lookup service rendering the service unavailable to other services in the network.

---

[167] See section 6.1.2.

[168] A piece of mobile code sent out by the Jini service.

[169] Lookup service: a central registry service on the Jini network that allows services to find each other. It can run anywhere in the network and can be federated with other lookup services. It provides registration, access, search and removal of services.

[170] A piece of mobile code sent out by the lookup service.

[171] A container for several objects, including an object called a service object, which clients can use to interact with the service. The service item can also include any number of attributes, which can be any object. Some potential attributes are icons, classes that provide GUIs for the service, and objects that give more information about the service.

The figure below gives a schematic overview of the events described above:

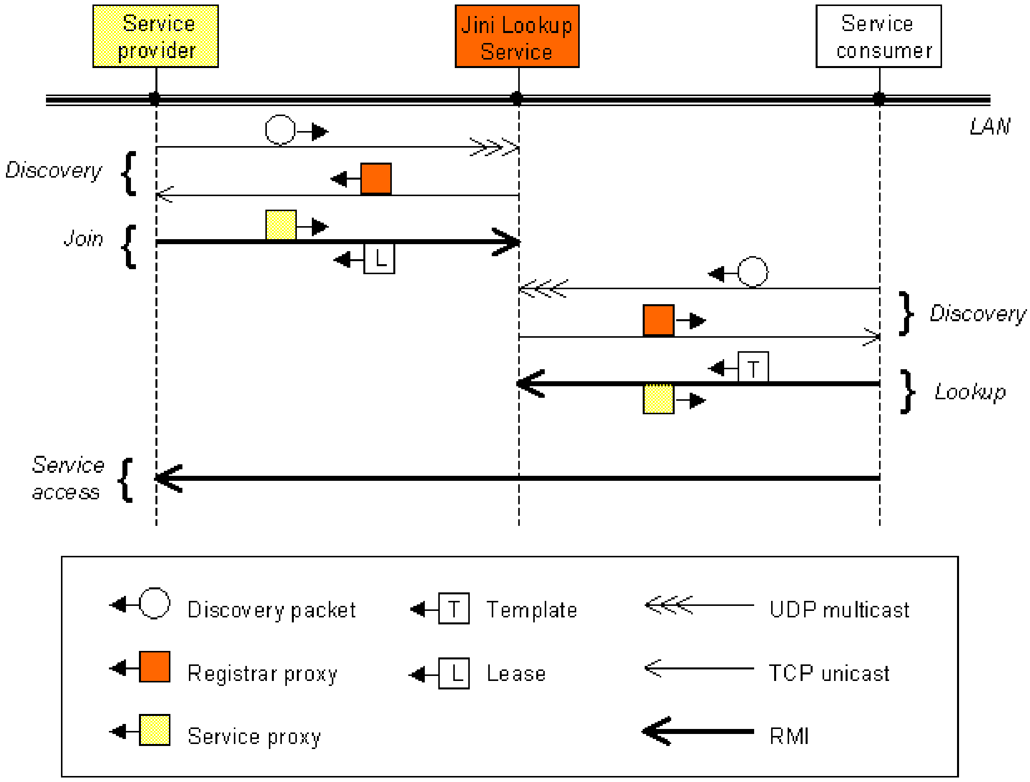



**Figure 6-4: Sequence diagram of a service joining the Jini network and a consumer finding and using it.**

Once a service is registered with the lookup service it can be found by other services, which will be its consumers. When a consumer searches for a service on the Jini network it must first discover the lookup service. The mechanism to accomplish this is the same used by service providers joining the network, with the exception that the registration object is used for doing a service lookup instead of for service registration. Sending the lookup service a template of the desired functionality accomplishes the lookup task. The template is (partly) based on a programmatic interface[172]. This interface can be implemented by a service, for example a printer implementing the printer interface. Consumers looking for printing functionality can in turn query the look up service for services that implement the printer interface. The lookup service

[172] A programmatic interface specifies the functionality a piece of software provides, not *how* it provides it. This allows for abstraction of different implementations of the same functionality.

will return all service items from its registry that match the template. The consumer receives the service item and uses it to communicate directly with the service provider.

This process ensures a networking system that is self-healing, has a low administrative overhead and is very scalable, all of which are qualities needed for this research. The disadvantage of using Jini is that it is Java based and thus requires every service in the network to be compatible with that language. However this is often not as problematic as it may appear as existing functionality can usually be “wrapped” in a “Java friendly” package. The advantages of Jini are such that it serves the purposes of this research — to implement the scenarios.

### 6.2.3. ServiceUI

The next decision concerns the user interface; the Services Platform needs to accommodate diverse interfaces. The need for such diversity is a consequence of the platform independence requirement because different hardware platforms support different interfaces. As was seen in the previous section dealing with Java, once such hardware platform independence is achieved, a wide range of devices can consume a service.

The need to support multiple diverse interfaces becomes clearer when considering an interface for controlling the lights in a house that can be accessed from both a mobile telephone and a desktop computer. Clearly the display capabilities of the telephone are significantly less than that of the desktop computer. Moreover the latter probably has the processing power needed for speech recognition[173], more screen space, better input devices, more advanced power management, and so on, all of which can help ensure a better user experience. Such differences between devices need to be minimised when an identical, or at least very similar, user experience of a service is required across platforms.

One approach to solving this problem is to provide an interface for all possible devices with each service. The obvious disadvantage to this approach is that it requires developers to create many interfaces for each service. This disadvantage however may be offset by the benefits of ensuring that users get the best possible interface with the service.

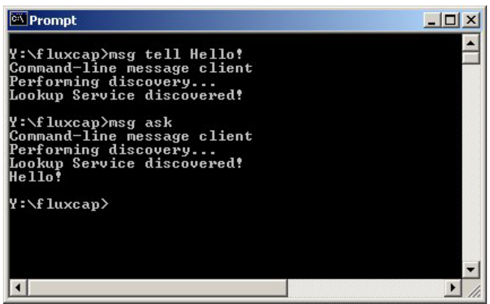
[173] At the time of this writing.

One way to attach interfaces to a Jini Service, is to use the ServiceUI[174] platform (Venners, 2002b). As can be seen below a ServiceUI is built on top of Jini:

| |
|---|
| **ServiceUI** |
| **Jini** |
| **Java** |
| **Operating system** |

**Figure 6-5: ServiceUI in the technology stack.**

The ServiceUI allows one service to provide a number of different interfaces. Clients requesting the interface can automatically pick the appropriate interface and display it without the need for user intervention. Such interfaces range from the familiar windows based interfaces, to speech interfaces, or HTML-based interfaces. The table below gives a few examples of interfaces for an instant messaging service developed for this research[175]. It should be pointed out that the table shows only a few interfaces, in a rapidly changing situation new possibilities, and limitations, come and go continuously.




←

This interface has no graphical components. It only consists of text in- and output through a command line interface.

**Figure 6-6: Command line interface.**

[174] An acronym of Service User Interface.

[175] Developed in close cooperation with Jakob Eg Larsen from the Technical University of Denmark.

➔

A PDA such as the one shown here has a little more than text capabilities. Its interface is in black and white and is almost completely text-based despite having some buttons and a few basic graphical interface components.

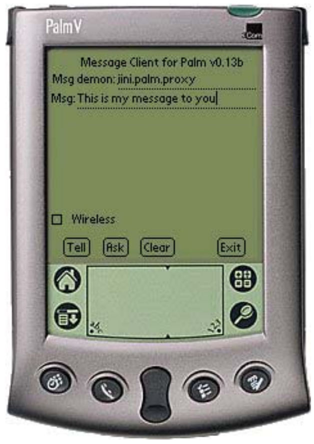



**Figure 6-7: PDA interface.**


**Figure 6-8: Large screen interface.**

🡐

A large screen that can be used for accessing the service has quite dissimilar characteristics to the other devices shown in this table.

It is often viewed from a distance.

Has poor text display capabilities.

➔

A fully functional windows-like interface. This example was built in Java using the Swing interface components library.

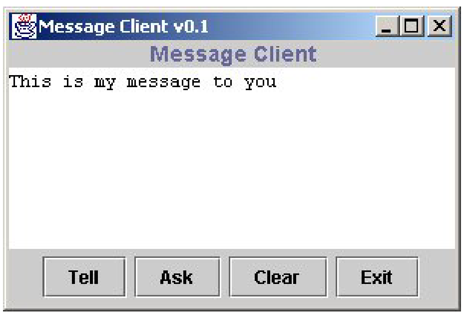



**Figure 6-9: Java Swing interface.**

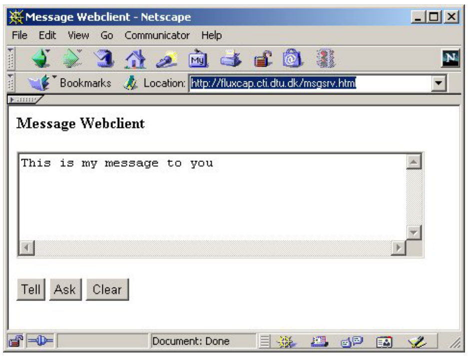



**Figure 6-10: Web interface.**

🡐

A Web-based interface has to be viewed using a Web browser. As HTML was not designed for building the complex interfaces needed for highly interactive services (Nielsen, 2000a), there are limitations to what you can do with it.

🡒

A WAP interface shown on a mobile phone. WAP on a mobile phone is severely limited in its interaction capabilities. At most it can show a couple of lines of text and an input field.


**Figure 6-11: Mobile phone interface.**

ServiceUI compatible interfaces require developers to clearly separate interface functionality from service functionality. The standard way to do this is by following the Model-View-Control design pattern[176]. An interface-service separation means that the ties between the interface and the underlying system are kept to the minimum, leaving more room for experimentation with different interfaces. One such possibility is to develop the interface in a different programming language than that used to develop the underlying system. This leaves developers the freedom to use the language best suited to each purpose.

---

[176] A design pattern in software design as described in (Gamma, 1995) as "descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context".

Another solution for associating multiple interfaces with a service could be to develop a language that describes the interface so that the device can construct the interface based purely on the description (Hodges and Katz, 1998). This avoids the problem of having to develop multiple interfaces for one service. There is already research going on such languages[177] but unfortunately these kinds of languages are not yet sufficiently mature to be of use for the purposes of this research.

### 6.2.4. Place API

The last level at which a decision needs to be made relates to the organisation of a collection of services. Just as the applications on a standalone desktop computer can be organised in folders by activity[178], or on the desktop by name, services can be organised by the service providers or consumers (see section 3.5 for a discussion about this). In some cases special technologies are needed to accomplish such an organisation, just as a file hierarchy is needed on a standalone desktop computer to organise applications by activity[179]. The technology that will be used to organise the prototype services is the Place API, which sits on top of the ServiceUI layer and completes the technology stack used for this research.

| |
|---|
| **Place** |
| **ServiceUI** |
| **Jini** |
| **Java** |
| **Operating system** |

**Figure 6-12: The Place API in the technology stack.**

The Place API is being developed by a small group of volunteers[180] as an open source project called the 'Place project[181]'. The goal of this project is to create software (the Place API (Venners, 2002a)) that allows for services to be organised in virtual places.

---

[177] See (Hodges and Katz, 1998) for instance.

[178] For instance office productivity tools organised in one folder relevant for work.

[179] Whether this file hierarchy is displayed as plain text, 2D graphics or a 3D world doesn't really matter, that's up to the 'mode layer' of the hierarchy of navigation.

[180] Including myself.

[181] http://cyberspace.jini.org/

A virtual place is a collection of links to services and/or virtual places. The fact that virtual places contain links both to other places and to other services means that users can navigate between places in search of services using appropriate services on a place-by-place basis. A place is itself a service, and any service can be, (or become) a place by offering links to other services. An example of a place might be a 'Home Place' offering links to services relevant to a home, such as a service for adjusting the heating or turning on the lights in the house.

The fact that the term 'place' was chosen does not of itself mean that users have to follow a geographical metaphor. Places can be created based upon many principles such as activity, time of use, name, or some other principle[182].

The use of a geographical metaphor neither forces service interface designers to provide an interface that mimics physical reality. As Dieberger showed (Dieberger, 1994) it is entirely possible for designers to use a city metaphor, in other words to use a geographical metaphor in text mode to create an interface[183].

The fact that the Place API fits well with the other technologies chosen for the Services Platform, and the freedom it allows to experiment with different ways of organising services, makes the Place API a useful technology for this research. Particularly as it will allow different implementations of the scenarios to be developed and tested.

## 6.3. Services Platform184

The fact that the requirements have been identified, and the choices of technology made, means that the Services Platform can be developed. In the pages that follow the specification of the Services Platform will be given, followed by an overview of the implementation.

### Specification

Standard client-server applications interchange data that needs to be interpreted both by client and server applications. This is similar to how the Web usage in which data is presented using a standard format that is interpreted by Web browsers and Web servers alike. In both cases the clients need to 'understand' the format of the data

---

[182] See 3.5 for more examples of organisational methods.

[183] Also see 3.7 for more modes.

[184] The Services Platform was developed in close cooperation with Jakob Eg Larsen. The work is described in (Larsen and Beute, 2001).

being sent over in order to process it properly. The Services Platform however relies on mobile code to accomplish its tasks[185].

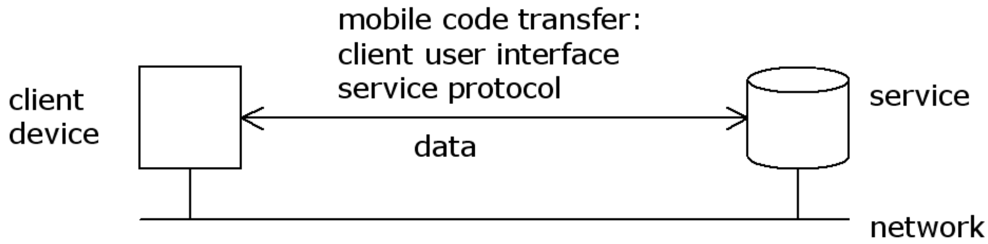



**Figure 6-13: The Services Platform model.**

Mobile code ensures that the client is capable of executing code sent by a service, thereby ensuring that the Services Platform has the possibility to transmitting functionality (behaviour) to clients that do not already possess it.

In order for a service to make itself available on the network through the Services Platform it must have the following components:

1. A service data protocol component (called a *proxy*).
2. One or more interfaces for different clients.
3. A service functionality component.

The Services Platform data protocol is therefore used to transmit a mobile code module, the *proxy,* when services are accessed from a client device. The *proxy* implements the service specific data protocol[186], which means, that unlike the traditional client-server model, clients no longer need to have prior knowledge of the data protocol (Waldo, 2000). In other words, the Services Platform does not require pre-installed applications on the client side. Once the *proxy* code has been sent to the client device, the client and the service can communicate using the *proxy*.

Given that a number of interfaces can be associated with a service the next step is to fetch an interface[187] for the specific device used to access the service. These interfaces are specifically tailored to each *service*, thereby ensuring optimal usability.

[185] As discussed in section 6.2.1.

[186] The *proxy* can implement the data protocol best suited for the functionality of the service, for instance using TCP, Remote Procedure Call (RPC), HTTP, or any other protocol, or a combination of these.

[187] See the ServiceUI technology discussed earlier in this chapter.

Furthermore each interface is tailored to *specific interface toolkits*[188], which means that a client device containing a particular toolkit has access to a fully functional interface[189]. As a consequence of this multiple client devices can be used to access the same service. Service interfaces are attached to the services, and a mobile code module that implements the interface is sent to the client device when a user requests to use the service. The interface is then rendered on the client side —typically on the client device's display, thereby enabling users to interact with the service back-end.

A flexible mechanism such as that described in the previous paragraph means that interfaces can be implemented using the interface mode most suitable to a particular service, be it a graphical interface, a hypertext-based interface, a speech interface, or some other appropriate mode. Nor are service interfaces limited to visual interfaces; the flexibility of the architecture enables implementations of any interface supported by the client device, and this can include speech interfaces, voice recognition, and handwriting recognition. The ability to support such diverse interfaces can be an important advantage, as interfaces are an important factor to the success of products and services (Myers, Hollan, and Cruz, 1996).

## The navigation layer

Access to the collection of services is provided through a navigation layer, which is the interface for accessing and organising services, see Figure 6-14. It is the core component of the task of coordinating the users' interactions with the collection of services and is the most important component for this thesis. The navigation layer is where the layers of the hierarchy of navigation[190] (for services) can be identified. It is in short a nexus, and has the same role in service provision and navigation as an operating system does in coordinating users' interactions with their desktop computers (Nielsen, 2000a), and Web browsers do for users accessing the Web. The navigation layer enables the user to access and organise services on the Internet. The design chosen for an interface is dependent upon the functional properties and form factors of the device for which it is designed. Thus an interface can have many possible designs. The whole purpose of the navigation layer is to give users access to all of their services irrespective of the device used to access the Internet. An example of this would be enabling a user to use a messaging service from a mobile phone, a PDA, and a desktop computer.

---

[188] A toolkit is an integrated set of software routines and utilities that are used to develop applications. An interface toolkit contains routines and utilities necessary for building an interface.

[189] Mobile code furthermore ensures that any missing components can be sent over the network to the client.

[190] As discussed in section 3.3

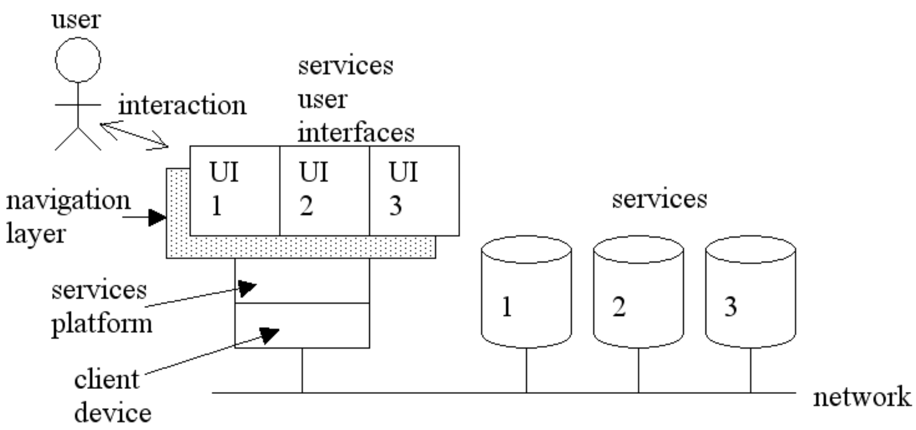



**Figure 6-14: Services Interfaces in the Services Platform.**

## Implementation

The Services Platform was implemented using the technologies chosen in section 6.2 of this chapter. Figure 6-12 shows the technology stack used to implement the Services Platform resulting in a prototype as shown in (the slightly more colourful) Figure 6 below. Both stacks start with the hardware platform running an operating system[191], which in turn run a Java Virtual Machine on top. The Jini platform sits on top of the Java Virtual Machine, and enables services to register on the network, and announce their capabilities; moreover it enables clients to lookup and access services. The navigation layer on top of Jini is implemented using ServiceUI and the Place API[192].

[191] Although the term 'operating system' is used mostly in connection with desktop computers, every appliance that contains a processor has an operating system. Whether it is a mobile phone or a Web server.

[192] See section 6.2.4.

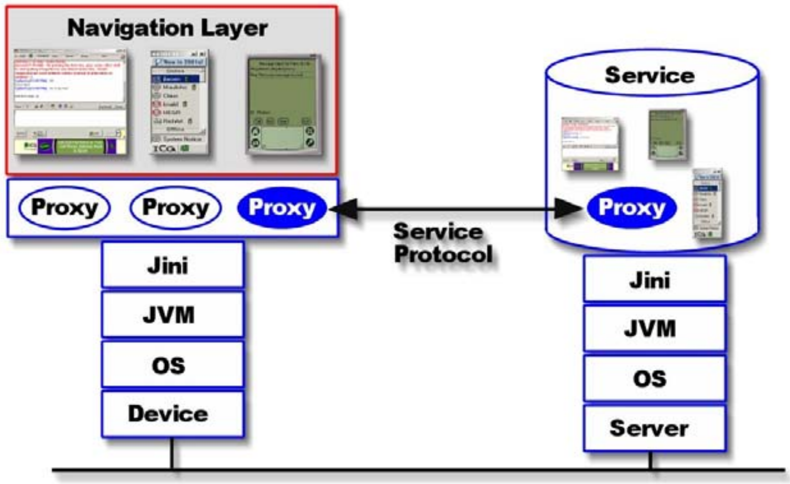



**Figure 6-15: Services Platform Architecture Overview.**

The architecture functions as follows; on the service, or server, side (right-hand stack in the figure) the service is registered with the *proxy* that implements the data protocol for client-service communication together with a number of interface objects for client platforms. At the client side (left-hand stack in the figure) there is the navigation layer that provides the user with an overview of available services. When a client accesses a service the service *proxy* and interface are downloaded, and the interface made shown on the client device.

## Security

The security mechanism in the Services Platform is adopted from Java and Jini technology, mobile code is run in a "sandbox"[193] that encapsulates the mobile code on the client side. The effect of this is that the mobile code is not allowed to access vital system resources, and is only permitted to connect back to its originating service. The security aspects of using mobile code are very important, but are beyond the scope of this paper[194].

The Services Platform has certain strengths that make it an appropriate platform for this research. First, unlike client-server applications it does not require installation or maintenance on the client side. Nor does it have to resort to the partial solution

[193] A restricted environment for software to run in where certain functions are prohibited.

[194] Readers interested in this topic can learn more about the Java security architecture in (Gong, 1997).

adopted for certain Web-based applications, in which a plug-ins is still downloaded and installed to the universal Web browser client to enable it to handle unknown content. Second, the Services Platform supports interfaces for several client devices and the interfaces are tailor-made for the various client devices. Third, it allows developers to create fully functional interfaces that overcome some of the problems with HTML-based interfaces. Finally, it enables the service provider to update the service transparently, without requiring any installation on client devices.

## 6.4. Summary

This chapter introduced the Services Platform. It first identified the requirements by using scenarios. Next, it explained why certain technologies were chosen to implement the Services Platform, and finally it described the Services Platform itself.

Given that the Services Platform is now in place it can be used to experiment with different implementations of the navigation layer, making it possible to appraise different combinations of the layers of the hierarchy of navigation. The result will be a wide range of ways to navigate services, which can be compared and tested for insight into the research topics. The following chapters describe and evaluate the results of such experimentation.

# 7. Implementations

*Having dealt with the issues arising from designing the Services Platform this chapter focuses on using the platform to explore different ways of navigating services. This requires different implementations of the navigational layer of the Services Platform*[195] *and the implementation of multiple services.*

Taken together the implementation of the Services Platform, the browsers and the services consist of over sixteen thousand lines of source code[196] representing the majority of work done for this thesis. Since most of the technical details are not directly relevant for the argumentation of the thesis, this chapter will only provide a high-level overview[197].

An implementation of the navigation layer of the Services Platform will henceforth be called 'browser'. Such a browser allows users to navigate services and should not be confused with Web browsers. This chapter first introduces the browser and its separate user interfaces. Next, it will give an overview of the services that were implemented and needed for implementing the scenario. This will be followed by a discussion about the way the services were organised and which metaphors were used. Finally, the different implementations of the scenarios will be described.

The variables when implementing the browser and services are specified by the hierarchy of navigation described in section 3.3. Choosing the organisation of the services, the metaphors to use, the mode of the interface, and the navigation technique to support, was rendered somewhat more complicated by the fact that these variables could be combined in many different ways. The solution chosen was to only implement those relevant to testing the hypotheses. Furthermore in order to deal with the subject matter of the second hypothesis (see chapter 5) I concentrated my efforts upon using a geographical metaphor.

When it came to interface design I tried to follow the guidelines available from the literature[198]. As the focus of this project is upon navigating *between* services as opposed to navigating *within the interface* of a particular service it would have been ineffective to spend too much time on interface design. To design usable interfaces

---

[195] See section 6.3.

[196] Approximately the same as 500 pages of text.

[197] For those interested in the low level technical details the source code is available from the CD-ROM that accompanies this thesis.

[198] I mainly used (Cooper, 1995;Dix, 1999;Laurel, 1995;Mayhew, 1992;Sutcliffe, 1995;Preece, Rogers, and Sharp, 2002;Nielsen, 1993) (Preece, 1994;Tognazzini, 1992).

within a limited timeframe was nevertheless necessary and I found the guidelines given by Tognazzini[199] to be of considerable assistance.

## 7.1. Browser

An explanation of the underlying mechanism of the browser will be followed by an overview of the different interfaces that were implemented.

### 7.1.1. High level overview

The browser is the only thing necessary for users to have installed on their devices to navigate services; it acts as a bootstrap mechanism[200]. When the browser starts, it searches the local area network within multicast range[201] for services. Once that is completed, any local services found are available to the user. The services can be divided into two categories; 'conventional services' and 'places'. A 'conventional service' is a service as defined in this thesis while a 'place' is a collection of links to other services or other collections of links[202]. Users can navigate by going from place to place and interact with the different services and places available. The services and places accessible from a place can be listed so that users can decide either to use a service or to navigate to another place. Assuming that a desired service has been found, and that it has an appropriate[203] user interface, it can be launched. The example session provided as an appendix[204] shows a user doing so.

---

[199] Tognazzini's 'First Principles' were particularly useful:

http://www.asktog.com/basics/firstPrinciples.html

[200] A bootstrap mechanism means a way to get started, in this case to get started navigating services by using the browser. Theoretically it is possible to make the browser itself a service, but that would in turn require another installed application to launch it.

[201] IP multicast is a way to transmit a message to multiple recipients at the same time. The range is delimited by the settings of the routers on the network. Usually the multicast range is considered to be the local area network.

[202] It should be pointed out that since a 'place' does not necessarily correspond to physical places the use of the term does not mean that this organisation only allows a geographical metaphor. However since a geographical organisation is a focus point of this research, and 'place' is a convenient name, the terminology seemed fitting. A place could, for example, contain all services relevant to text editing, which would make it an activity-based organisation.

[203] One that can be rendered by the platform the browser runs on.

[204] Section 11.1.

### 7.1.2. Use case

To clearly describe the general characteristics of the browser a *use case*[205] is appropriate.

Irrespective of whether a service is a hardware or software provided[206] service the use case diagram is the same for every version of the browser:

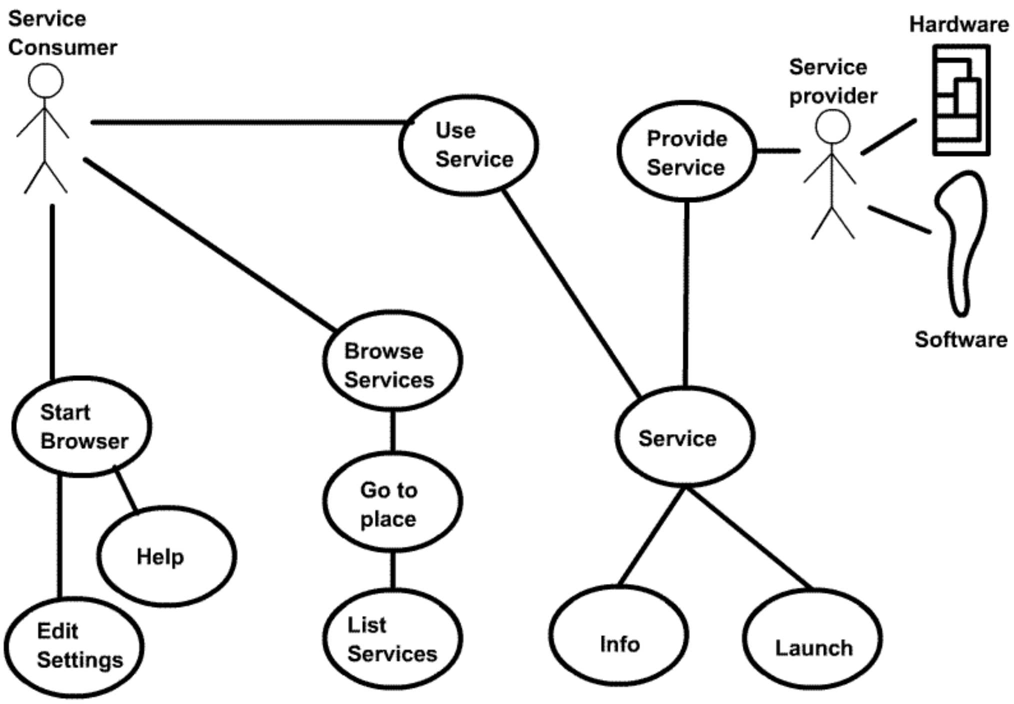



**Figure 7-1: Use case diagram.**

1. A user starts the browser and has the option of either changing some settings or of asking for help.
2. Once the browser is in operation the user can browse for services, either by directly *using* a service or by *going to* a service (if that service is a place — that is it contains links to other services).

---

[205] A use case is a modeling technique used to describe what a new system should do (or what an existing system already does) from the user's point of view.

[206] Services can be offered both by hardware or software. An example of the former from the 'Travelling photographs' scenario is a television offering the service of displaying images, while an example of the latter from the same scenario could be an image editing tool service offering image-retouching functionality.

3. Once the user finds the required service, they can either launch a user interface to it or request further information.

### 7.1.3. Sequence diagram

The sequence diagram below shows the sequential interactions between the separate parts of the system as follows:

1. The user starts the browser.
2. The browser creates a data-model to keep track of all local services.
3. The browser updates the user interface each time new services appear or old ones disappear. Through the interface the user can not only view but also control the data-model.

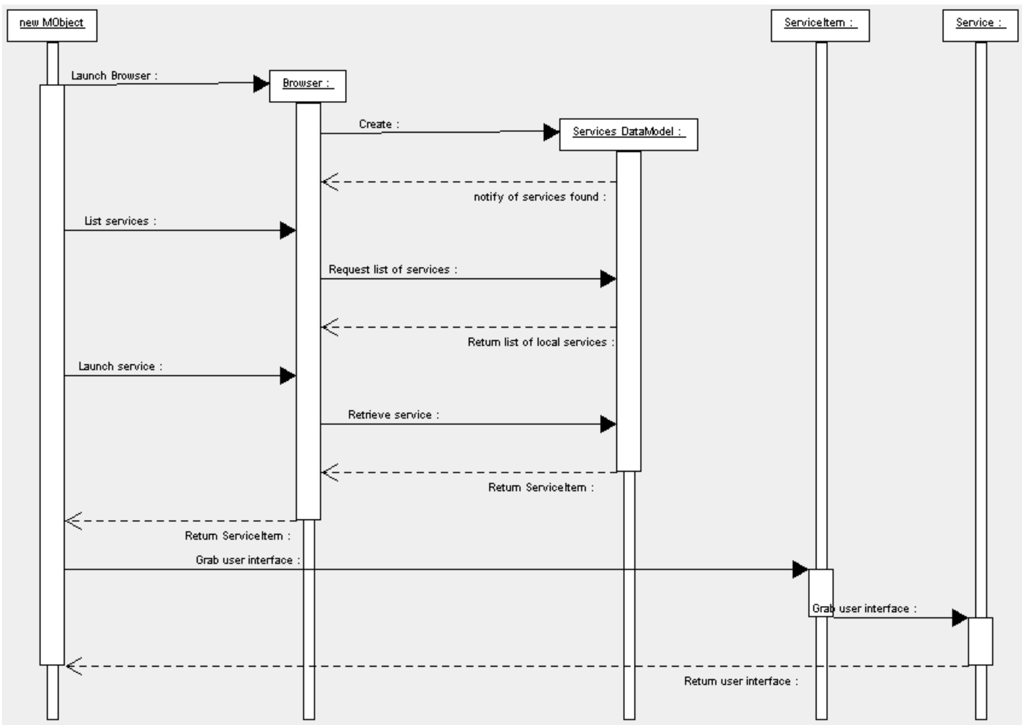



**Figure 7-2: A sequence diagram with time along the vertical axis (top is t=0).**

The user interface and data-model are updated using the model-view-controller paradigm (MVC)[207]. In the MVC paradigm the modelling of the external world, the user input, and the (visual) feedback to the user are explicitly separated and handled by three types of object. The *view* manages the graphical and/or textual output to the display. The *controller* interprets the inputs from the user (through mouse, keyboard, microphone etcetera), commanding the model and/or the view to change as appropriate. Finally, the *model* manages the behaviour and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller).

The MVC diagram shown below illustrates what happens when users find a service in which they are interested. There are two possibilities:

1. They can request the interface from the service directly.
2. Or if the service is a place the user can choose to navigate to that place.

Where the service is in fact a place the user instructs the browser (using the 'controller' object) to update the data-model with services in the new place. Subsequently the user will can view the new services using the 'view' object.

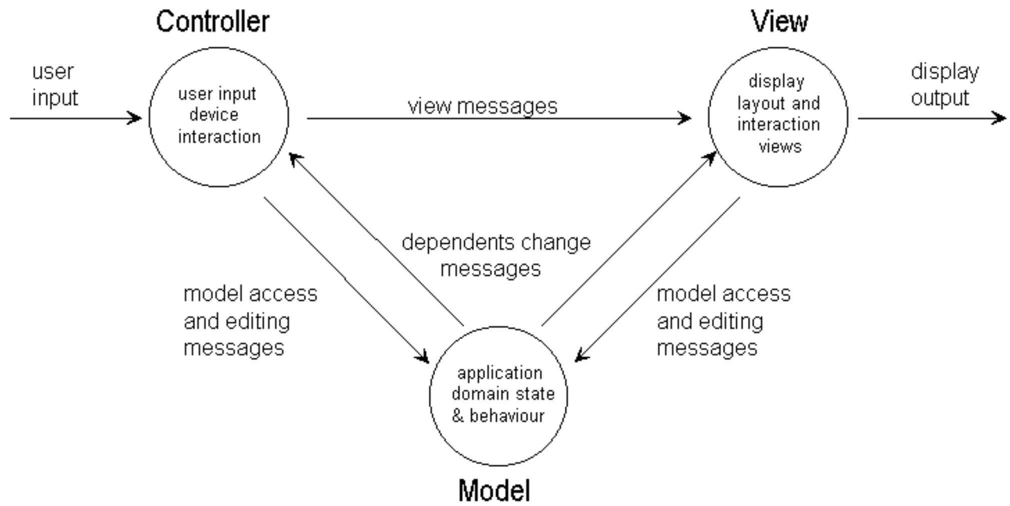



**Figure 7-3: Model-View-Controller (taken from (Dodani, Hughes, and Moshell, 2002, 255-262)).**

[207] The model-view-controller paradigm was first introduced with Xerox PARC's SmallTalk-80 interface and later generalised as a design pattern for software development (Gamma, 1995).

Once the model and controller are implemented they can largely be reused when experimenting with different interfaces (views). This will shorten the engineering time considerably.

### 7.1.4. Class diagram

In object-oriented software development objects are used as the building blocks for applications. Objects are independent program modules written in object-oriented programming languages[208], they can extend other objects inheriting its functionality[209]. All implementation for this thesis was done using object-oriented technology[210]. The browser can therefore be displayed as a class[211] diagram, which is a high level view of the class hierarchy. The diagram below only displays the most relevant classes, giving a summary of the browser's design.

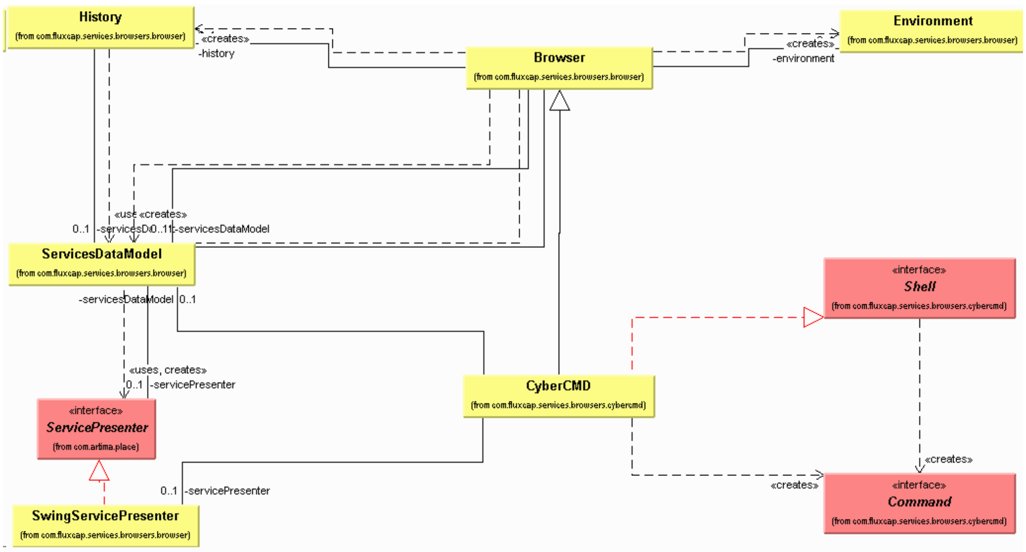



**Figure 7-4 Class diagram of CyberCMD.**

As the diagram shows the central class is Browser, which takes care of general behaviour such as tracking local services (through the ServicesDataModel class). It

---

[208] Example programming languages are Java, C++ and Python.

[209] An example would be the objects 'Triangle' and 'Circle' that both extend the object 'Figure' thereby inheriting all traits of a figure and extending it with particular traits.

[210] It is beyond the scope of this thesis to explain object-oriented software development in more detail, for more information see (Coad and Yourdon, 1990;Gamma, 1995).

[211] An object is an instantiation of a 'class'; therefore the term 'class' is used during design while the term 'object' is used during implementation.

also tracks services in the current place and updates the history *via* the History class. This class is kept as general as possible to avoid having to re-implement the same functionality for different implementations of the browser. Specific implementations of the browser, such as the command line version CyberCMD, extend the Browser class and inherit its functionality. Any implementation specific functionality is taken care of by classes associated with CyberCMD, but the Browser class provides most functionality, as CyberCMD is merely a browser extension geared towards command line interfaces.

The Browser class is also responsible for general settings such as the timeout for retrieving services (through the Environment class). CyberCMD handles those issues relevant to command line browsers, for the other browser interfaces the CyberCMD class and its associated classes were replaced with other classes. CyberCMD implements the Shell[212] interface[213], which means it fulfils the contract of a command line interface. It also creates many different instances (not shown in the diagram) of the Command interfaces such as the 'list', 'go', 'set', and 'launch' commands.

The use of programmatic interfaces ensures a flexible architecture by specifying the functionality and leaving the implementation to the individual implementers. An important interface is the ServicePresenter, which takes care of displaying the graphical user interface (or outputting the speech interface etcetera). Depending on the environment from which the browser is started it creates the appropriate ServicePresenter. In the class diagram a SwingServicePresenter is shown, which means a Java Swing[214] user interface will be rendered if the service provides one. As discussed previously a service can have multiple user interfaces associated. Other ServicePresenters are possible, e.g. a HTMLServicePresenter that renders user interfaces made in HTML. Which ServicePresenter is used depends upon the possibilities of the platform upon which the browser is running. Thus if it is running on a mobile phone, the requirements are more restricted than when it is running on a desktop computer.

### 7.1.5. Interfaces

To experiment with different ways of navigating the Services Sphere it is necessary to experiment with different user interfaces. Four types of user interfaces, or navigational modes in terms of the hierarchy of navigation, were implemented. These were:

---

[212] 'Shell' is the historically used name for command line interfaces.

[213] Programming interface, which is different from a user interface. It provides the form of a class without implementation. Classes that implement the interface have to implement the functionality. This a way to establish a protocol between classes (see (Eckel, 2000)).

[214] A windowing toolkit that is part of the Java 2 Standard and Enterprise Editions, but not of the Java 2 Micro Edition. The latter is aimed at small devices that most of the time have very limited displaying capabilities.

- A text interface.
- A 2D interface.
- A 3D interface.
- A speech interface.

I discuss each interface below and explain why I chose to implement it.

## CyberCMD – A text interface

CyberCMD is the initial user interface implementation for experimenting with navigation of services. It is a text-based (command line) interface that can be started in a shell[215] on any operating system that supports Java.

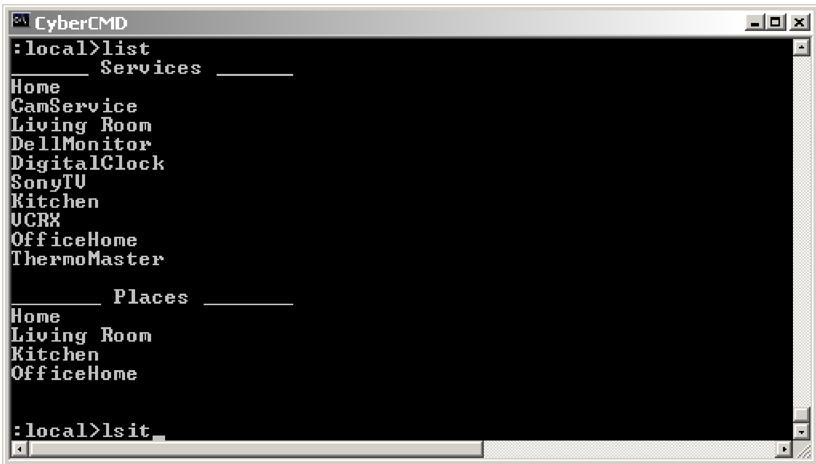



**Figure 7-5: A text-based user interface for the browser.**

When the CyberCMD starts it searches the local area network within multicast range for services. Once that is done a ‘list’ command shows the local services it has found. It then divides the services into services and places. The user can change places using the ‘go’ command and, assuming that the ‘go’ command completed successfully, the prompt will be extended with the name of the new place. Issuing a ‘list’ command again will show a list of services and places that are accessible from there. In this manner a user can go from place to place searching for services. Once the requisite

[215] Text based input mechanism for an operating system.

service is found and assuming it has an appropriate[216] user interface it can be launched. The example session provided as an appendix[217] shows a user doing so.

As Benyon points out (Benyon, 1995), the effectiveness of a text-based interface for navigating Information Spaces is highly dependent upon the spatial ability of the user. The effect of spatial ability on task completion time was most dramatic with command interfaces (which are text-based) when such interfaces were compared with menu, icon or button centric interfaces[218]. Task completion time with menu driven interfaces for instance were not affected by the subject's spatial ability. The question is whether the same applies to navigating the Services Sphere. Unfortunately, due to time limitations, quantitative user testing to clarify this issue is not possible within the scope of this thesis, but qualitative user testing with CyberCMD should point out whether a text-based interface is a promising way for navigating the Services Sphere.

## CyberMap – A 2D interface

Next a 2D interface, called CyberMap, was developed. There are many possible variations in 2D interfaces, of which the interfaces at present used on desktop computers are an example. Such an interface typically has tree structures, overlapping windows, and lists. However exploring all possibilities offered by such an interface would be beyond the scope of this research and thus only one variation has been implemented. It consists of a main window displaying all places and services of one place. As can be seen from the illustration below it contains a window showing:

1. A list of all places available.
2. A list of all services available.

---

[216] One that can be rendered by the platform CyberCMD runs on.

[217] See section 11.1.

[218] Benyon's terminology (Benyon, 1995).

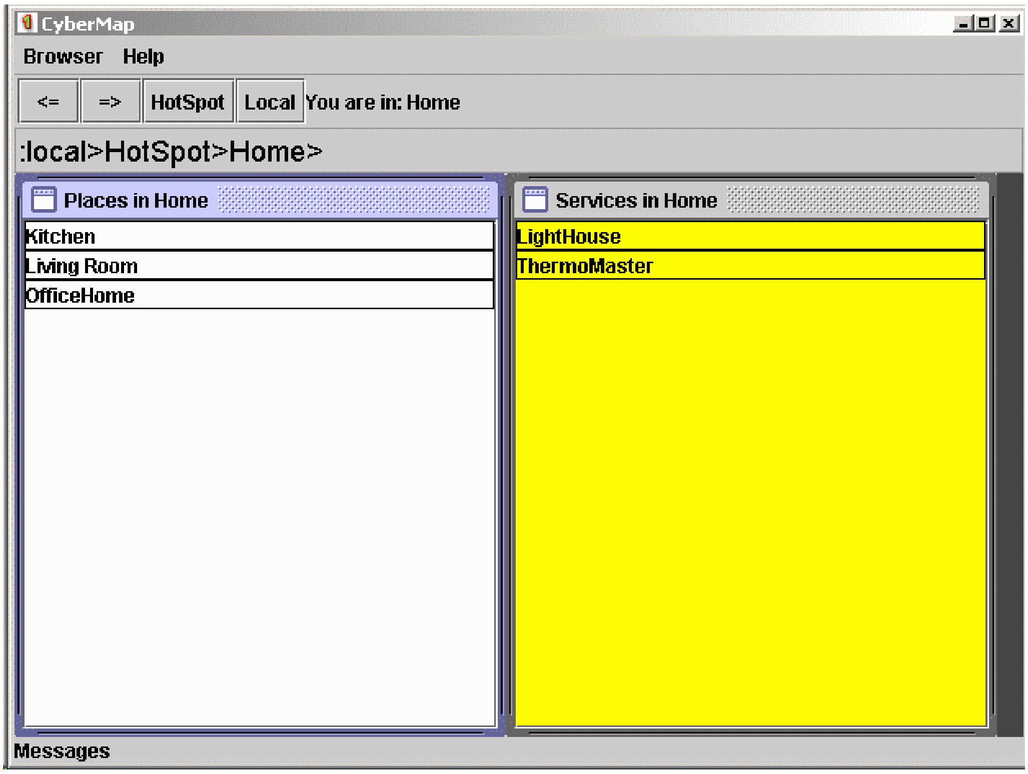



**Figure 7-6: A 2D interface for the browser.**

By clicking on the name of a place the user is taken to the new place and the internal windows are updated with the places and services of that new place. By clicking on the name of a service the interface for the service is retrieved and displayed to the user. Furthermore users can use the back and forward buttons for navigating through their history. For convenience there are buttons that take the user directly to their HotSpot (the place containing their favourite services) or to the local network.

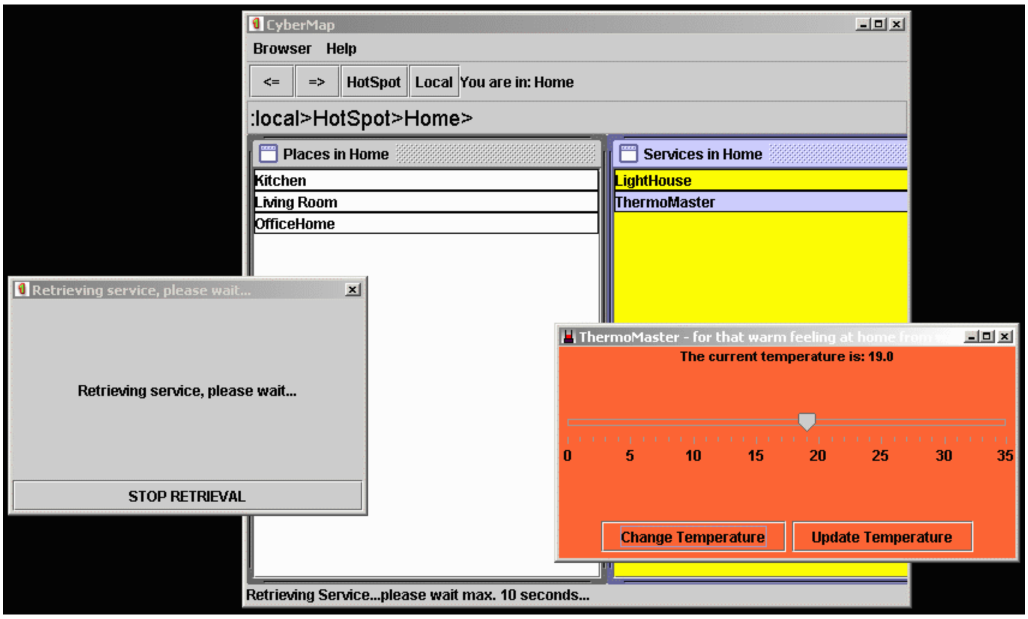



**Figure 7-7: Retrieving the interface for controlling the house's thermostat.**

There has been much has research into the effectiveness of 2D interfaces for many types of computer related tasks[219], such as Web browsing[220] or information retrieval (Hu, Ma, and Chau, 1999, 125-143). This research tries to find out whether such interfaces are also promising for navigating the Services Sphere.

## CyberSpeak – A Speech interface

Using IBM ViaVoice[221] and the Java Speech API (Sun Microsystems, 2001) a speech interface (CyberSpeak) was also implemented. By saying aloud the commands that they would normally either type in a text-based interface, or click in a 2D interface, users can navigate without using their eyes and hands.

Although both speech recognition and speech interfaces have been promoted for many years it is still unclear how useful this technology is from a usability point of view. In a recent online interview (Walker, 2002, E01) Ben Schneiderman from the Human-

---

[219] The majority of research in Human-Computer Interaction serves as an example (see (Preece, 1994) for example).

[220] Most of the work by Nielsen (Nielsen, 1993) for example.

[221] Commercial speech recognition software that is no longer on the market due to limited success amongst consumers.

Computer Interaction lab of the University of Maryland expressed this doubt about speech interface by talking about the latest research results of his lab;

> *"Researchers in [Ben Schneiderman's] computer science lab discovered through controlled experiments that when you tell your computer to 'page down' or 'italicise that word' by speaking aloud, you're gobbling up precious chunks of memory -- leaving you with little brainpower to focus on the task at hand. It's easier to type or click a mouse while thinking about something else because hand-eye coordination uses a different part of the brain"*
>
> *- (Walker, 2002, E01).*

The problem however, is that in some cases users may have no choice as to what kind of interface they use, if, for example somebody were to try using a service when driving a car they would have to keep their eyes on the road and hands on the steering wheel. For situations such as this speech interfaces are the only option[222], and it would be useful to know whether such an interface is a fruitful approach to navigating services.

## Cyber3D – A 3D interface

A 3D (sometimes called 'virtual reality') interface was partially implemented. The interface is functional in that it enables the user to navigate a virtual world filled with services, but it is not connected to a real network. In other words it is a horizontal prototype of the type mentioned in section 5.2.1. This horizontal prototyping approach was adopted due to the lack of time as constructing such interfaces is very time-consuming. The reason a 3D interface was implemented at all is to allow experimentation and to ask experts and users for their opinions on it. If the reactions are positive a fully functional interface (vertical prototype) could be developed for future research. What follows are a few screenshots to make clear what the virtual environment looks like.

---

[222] Although it is even questionable whether speech interfaces are acceptable in such situations. Recent discussions in the Netherlands about the use of mobile phones while driving highlight the fact that talking might occupy cognitive functions needed for driving.


**Figure 7-8: Using a 3D interface to navigate the services in a home environment.**

**Figure 7-9: The office.**

**Figure 7-10: The bathroom.**

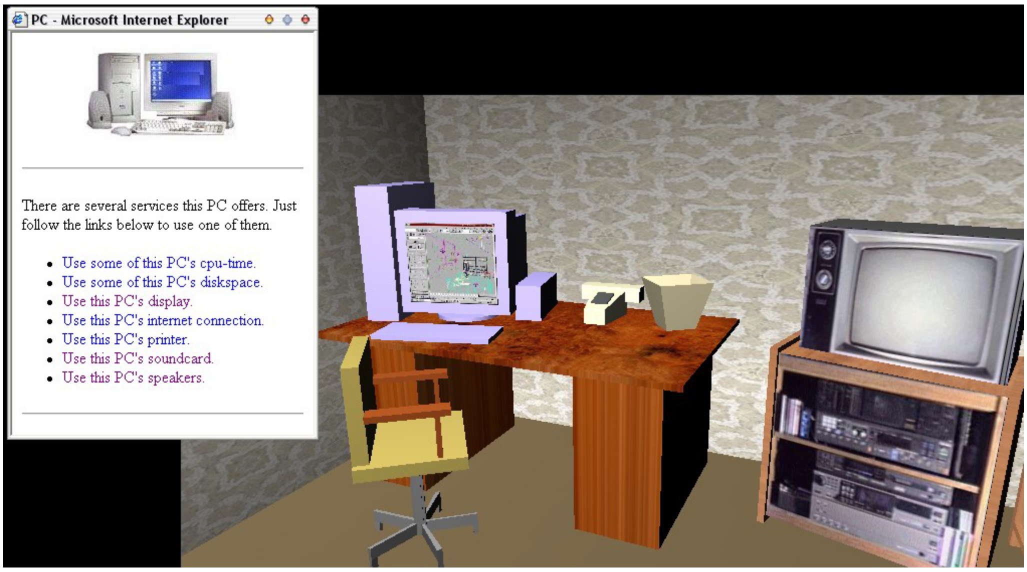



**Figure 7-11: Controlling the computer, using a combination of a 2D and a 3D interface.**

Virtual Reality interfaces have been researched extensively particularly during the early nineteen nineties (Steed, 1993) (for more recent research see (Card, Mackinlay, and Shneiderman, 1999)). Notwithstanding that research effort Virtual Reality interfaces have not yet entered the mainstream. The reason for this could be that most of the interfaces developed at that time were developed for navigating information, however as was seen in section 3.5, most collections of information (e.g. the Web) lack an intrinsic structure making it hard to superimpose an external structure (Dieberger, 1995). The fact that the interface discussed here is meant for navigating the Services Sphere, which might have an innate structure, could present a new opportunity.

## 7.2. Services

To create a convincing and usable test environment, a relatively large number of services, close to thirty, had to be implemented. Each of these services was relevant for the implementation of the scenario. The scenario will be discussed in the next section. First several of the services will be discussed. The purpose of this is not only to draw attention to the different issues each service addresses, but also to give a general idea of what is being referred to when talking about services. With few exceptions each service is provided with only one interface. In theory each could have provided multiple[223] interfaces, but time limitations made attempting to do so impossible in practical terms. — Nor would such an effort been worthwhile in terms of testing the hypothesis for validity.

[223] For instance a command line, pda, touchscreen, Web and WAP interface. See section 6.2.3 for examples.

## Camera


**Figure 7-12: The user interface of the cam service.**

The camera[224] is an example of a service provided by hardware. There are two camera services in the test environment one of which is situated in a home, and one of which is located in an office (see Figure 7-17). Apart from their locations the cameras are indistinguishable. The communication between the interface (displayed on the user's device) and the physical camera consists of a scheduled request by the interface for a new image. The camera then uploads the image using a service specific protocol[225]. It should also be pointed out that the interface is comprised of images and thus requires a platform that can handle such content. A desktop computer or the latest generation PDA's could do so, but not a monochrome phone interface.

---

[224] The term Webcam could have been used but would have clashed with the important idea that this camera does not rely on the Web but on the Internet.

[225] See (Waldo, 2000) for an explanation how application specific protocols can be accomplished. The music service as discussed in chapter 6 also uses a protocol that best fits its needs. It connects to a music file residing on a server somewhere and opens a streaming connection that allows the file to be played while it is still downloading. Streaming is a technology that is also widely used for viewing online videos.

## AgentX

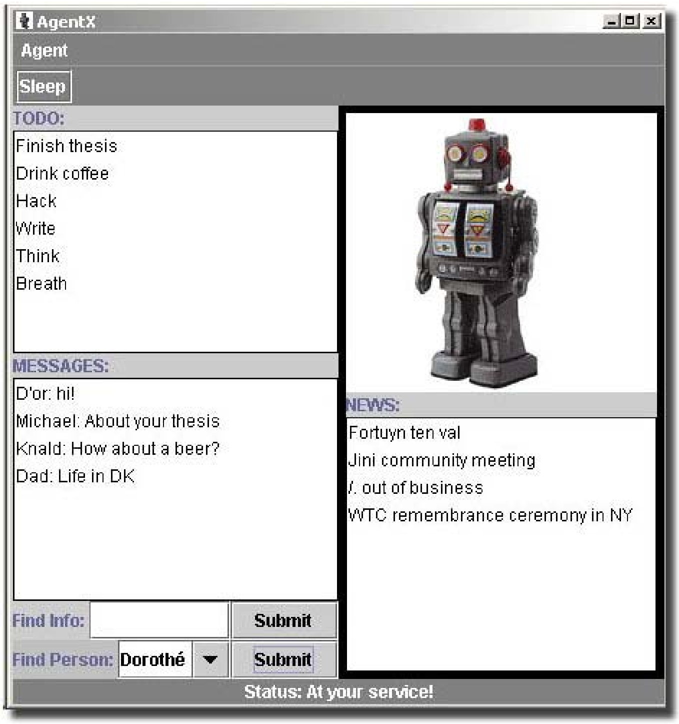



**Figure 7-13: The user interface of the AgentX service.**

The idea behind AgentX is to provide a personal agent service to which the user can give a small task, such as searching for information on a topic or a person, and send it on its way. Once the service has found information it considers relevant it returns with that information. It also provides news updates, e-mails, and a to-do list for the user.

The unique aspect of this service is that the agent's 'intelligence' does not *have* to be on the client side – making it a 'lightweight[226]' service. Such client side intelligence can be built in; the developer is free to determine the correct amount of client side intelligence to build in.

Only a small subset of the functionality was implemented for the prototype, but it has a fully functional interface, which is built using Java's Swing package. Swing is a standard part of Java for creating interfaces and is available only for standard installations of Java. It is thus not compatible with for example telephones or PDA's

[226] Lightweight services are services for which the client does not have to download large files in order to use them.

that run Java as the Java installations for such devices are smaller than the standard installation[227].

## JLog

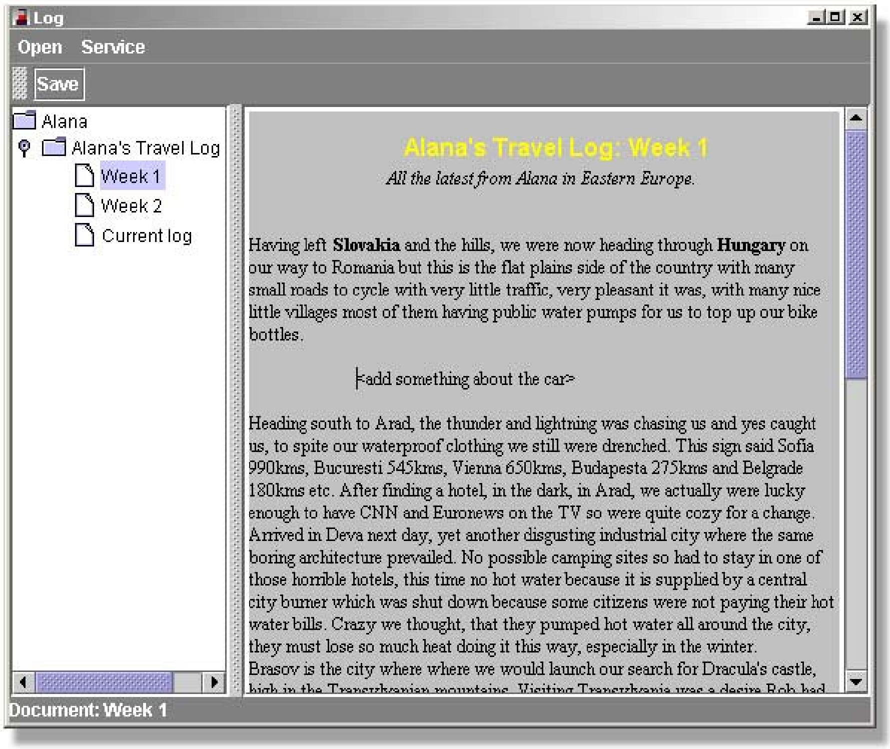



**Figure 7-14: JLog's user interface.**

Jlog is a service for maintaining an online log. Users can launch the service and start editing by choosing a log from the tree structure in the left pane. When they have finished an editing session the user can save the log and close the service by closing the interface[228]. The log itself will be saved on a Web server making it available as a Web page for others to read using regular Web browsers.

An important aspect of this service is that the log itself is stored 'server side' by the service. The fact that the service provider, who owns the server, handles the storage means that users do not have to concern themselves about such matters as, where their

---

[227] To be able to access the service from one of the latter platforms the provider could simply add an interface for that platform to the service.

[228] The interface to JLog was built using Java Swing.

log is stored, whether it has been backed up, or where to find the log the next time they want to edit it. They simply launch the service and start editing their log.

**TP**

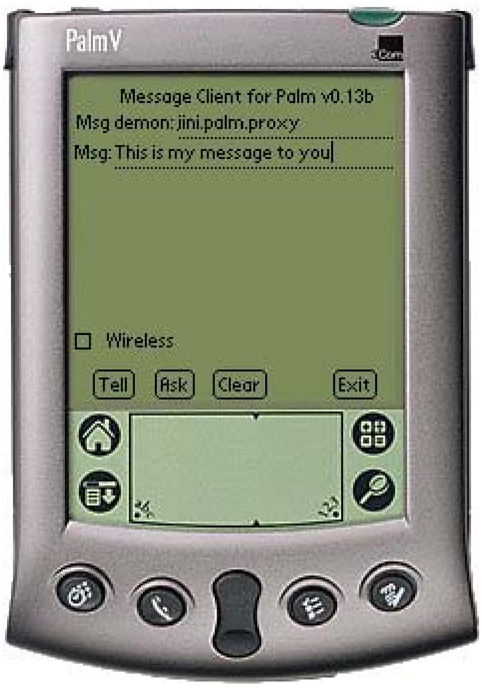



**Figure 7-15: The PDA interface of TP.**

TP[229] is a platform independent instant messaging[230] service. Its purpose is to provide a buffer between the users and the underlying technology thereby letting them focus on communicating with each without being concerned about the underlying technical details. This goal can be reached by removing both the limitations of the message format and limitations in accessing the service. Depending on the formats usable by the users' devices the first limitation can be solved by giving users the choice of whether they want to enter or consume a message in text format, as *braille*, or as a

---

[229] Developed by me in close cooperation with Jakob Eg Larsen of the Technical University of Denmark.

[230] A keyboard chat over the Internet between two or more people. Instant messaging is not a dial-up system like a telephone; both parties must be online at the same time. The names of people with whom a user wants to use instant messages is entered into a list, and, when any of those individuals log on, the user is "instantly" notified and can begin an interactive chat session. AOL's Instant Messenger (AIM), Microsoft Messenger, ICQ and Yahoo! Messenger are the major instant messaging services.

spoken message. Multiple user interfaces can be used to solve the second issue. (The various interfaces connected with this service that were implemented were presented in section 6.2.3). Having such choices available ensures that users are not confined to a desktop computer for using the service. They can equally easily access it from a PDA, a mobile phone, a television, a touch screen or any other platform for which the service offers an interface.

## Music player

The music service provides the user with access to music content over the Internet and play it on the client's side. The screenshot below shows one of the user interfaces:

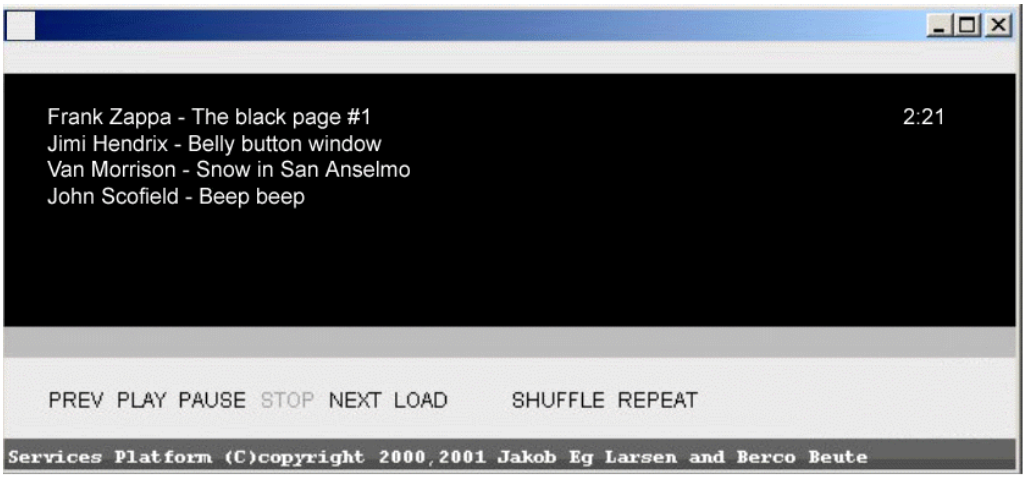



**Figure 7-16: Screenshot of the user interface of the music service.**

Using the graphical user interface the user can access music tracks and play them. An interesting aspect of this service is that the music format[231] is hidden from the user. Because the Services Platform enables mobile code to be sent over the network, code modules for decoding the music content are sent over the network as needed. For example, if a user were to select a track encoded using the MP3 format, and no client side code for decoding that format was present on the device being used, then the relevant decoder module would be automatically sent to the device and the track would play. This also allows for new formats to be used without the user having to install the proper decoder.

[231] The way the music is encoded, which can be MP3, WAV or another encoding scheme.

**Places: Home, Kitchen, Living Room & OfficeHome**

The services that link to other services are called 'places'. Any service can be a place, e.g. a car that links to a traffic information service, but sometimes a service does nothing more than to provide links. A couple of such services were developed for the test environment. There was a service called ‘Home’ that linked to other places such as ‘Kitchen, ‘Living Room,’ and ‘OfficeHome,’ and also to services such as the thermostat service (ThermoMaster). Whether a place possesses its own user interface is implementation dependent.

Which services reside in a place is predefined for the test set-up, meaning that the users can’t organise the services themselves. In theory there is nothing preventing them from doing so, but implementing that functionality would have taken too much time while it is not necessary for the scenario. I leave that for further research.

### 7.3. Organisation & metaphor

Now that both a collection of services and a browser supplied with different interfaces are ready it is expedient to examine how to organise the services at this point. Organising services can potentially be done by service providers or by service consumers. Service providers can organise their services after which the service consumers, using a browser, only have to render the collection of services. Service providers might also delegate the organisation to the service consumers; an example would be a service browser that organises an unorganised collection of services[232]. The approach taken here is to have the service providers organise their services (which doesn’t prevent the services consumers to superimpose another organisation).

As was seen when the hierarchy of navigation[233] was examined services can be organised in several different ways, by activity, by time, by name, or using some other criterion. Given that a goal of this research is to determine whether a geographical metaphor is more useful within a Services Sphere then it was within Information Space[234] it was decided to focus upon such a metaphor.

Accordingly the services are organised in virtual places using a geographical metaphor, in other words the virtual places corresponded to real geographical places and the services were accessible from the virtual place most natural for it. This raises the issue that not every service has an immediately obvious connection with a physical place. While for a television offering a displaying service it is obvious that the latter

---

[232] This is similar to a Web browser that displays a hierarchy of folders containing bookmarks. The Web pages (represented by the bookmarks) are not organised on the Web the way they are within the Web browser of the user.

[233] See section 3.3.

[234] Section 3.3 explained how close these two are connected.

has a connection with the room in which the television is physically located, it is less obvious for something as abstract as a text editing service. This point is important when using a geographical metaphor to enhance user understanding of the Services Sphere since such services are harder to incorporate in the metaphor. I shall return to this issue in the following chapter.

## 7.4. Scenarios

The platform, browser and services were developed (and the organisation and metaphor chosen) to implement the scenarios given at the beginning of this thesis. This section will show how the 'Travelling Pictures' scenario can be implemented using the browser and services. The text-based interface for the browser is used[235] for the purposes of the examples

To create a convincing test environment it is essential to have sufficient services and places distributed across the Internet. To meet this criterion the configuration shown in the illustration below was created.

---

[235] Interspersing pieces of text instead of screenshots of the 2/3D interface prevents cluttering of the pages.

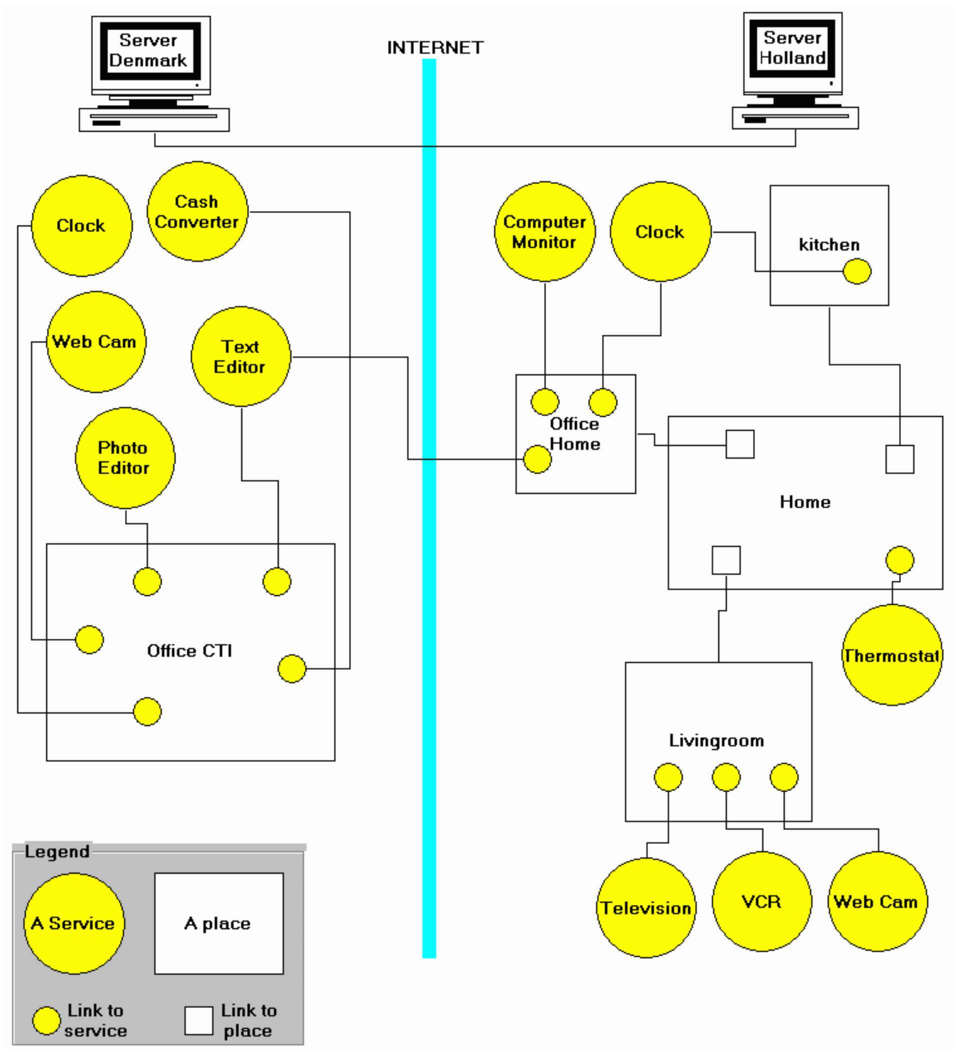



**Figure 7-17: Services environment.**

Several points need to be made here. First, in the interests of clarity not all of the services and places that were implemented are shown in the diagram. Secondly, the services are running on different servers on the Internet. Moreover the term 'server' should be interpreted as broadly as possible as anything from a wristwatch to a supercomputer can provide a service. Third, at the time this research was conducted there were no devices available that could provide the service themselves so the tasks were in fact 'performed' by computers. Thus the thermostat service (ThermoMaster) was not provided by an actual thermostat, changing its settings did not change a real thermostat but it altered a value on a server providing the service. This seeming limitation is in fact of no importance to the purposes of this research as the user experience was nearly identical[236]. Finally it should be pointed out that 'normal' servers would provide services such as the text editing service, so no 'workaround' was needed for those services.

The diagram shows services on a server both in Denmark and the Netherlands. Some of these services are places while others are 'conventional' services. As can be seen a link in a place can be to either another place or to a service. Such services do not have to be located on the same local network. They can reside anywhere on the Internet. In the interest of keeping the diagram simple only one such 'remote link', that for the Text Editor, has been included although such links occurred more regularly in the test set-up. The effect of such remote linkage is that the user uses the same service irrespective of the (virtual) place from which they launch it[237]. A different situation arises when a device such as a Web camera is providing the service (WebCam). Since actual Web cameras are providing this service the location of the camera makes a difference, whether it is the camera in the office at CTI, or the camera in the living room of home.

## Browser usage

When a user launches the browser on a network that is local to one of the servers, for example the one in the Netherlands, the sequence of activity is as follows (see section 7.1 for a more thorough discussion of this sequence):

1. When the browser starts it locates all services residing on the local network (which are services provided by servers[238] on the local network, including the one with the services as in Figure 7-17).
2. The user has the option of starting services or, if the service is a place, of navigating to that place.
3. From within a place the user can navigate to other places (possibly outside the local network) and use services available from there.

When the browser is launched on a network where no local services are available the sequence of activity is somewhat different. In that case the browser gives the user the option of navigating to those places, or launching those services, for which the user has created a link in his personal set of links. Such a set of personal links is referred to as the user's 'HotSpot'. Each user has a HotSpot containing links to their favourite

[236] The experience would have been different for the user if she was in the room whose heating was adjusted.

[237] In the diagram the service can be launched from a place called 'Office Home' and a place called 'Office CTI'.

[238] Where, again, 'server' should be interpreted broadly, it includes all entities providing services (as defined by at the beginning of this thesis).

places and services, somewhat analogous to the bookmarks seen in current Web browsers.

In the test situation it was desirable that all relevant services were available to the user, the HotSpot therefore contains a link both to the Home place and to the Office CTI place. This ensures that all services relevant to the test are accessible to the user even if the browser is launched outside the local[239] network.

## Scenario: give slideshow

In the scenario the actor connected a digital camera to the network in order to give a slideshow. The actor made the following two decisions:

1. To retouch some of the pictures before giving the slideshow.
2. To give the slideshow on the television.

Two issues (or problems) follow from those decisions:

1. A photo editing service needs to be found and used to edit some of the images stored on the camera.
2. The display service offered by the television has to be located and sent the images.

Both of these tasks are to be accomplished from within the room in which the actor is located.

The following is a step-by-step description of how these tasks can be accomplished using the browser and the services provided:

1. The actor connects the camera to the network

   *The camera automatically registers itself with the local network and announces its service, which provides content (images) to other services.*

2. The actor decides to access the camera's user interface using a computer on the local area network, so she launches the browser.

   *The browser finds all of the services available on the local network, including the digital camera.*

[239] Where local means the same network as where the services were running.

3. The actor types ‘list’ to see all available services.

```
:local>list
______ Services ______
Home
CamService
Living Room
DellMonitor
DigitalClock
DigiCam
SonyTV
Kitchen
VCRX
OfficeHome
ThermoMaster

_______ Places _______
Home
Living Room
Kitchen
OfficeHome
```

**Figure 7-18: Screen output of CyberCMD.**

4. The actor locates the camera and launches its interface by issuing a ‘launch DigiCam’ command.

   *The following user interface appears*:


**Figure 7-19: User interface for the Digital Camera.**

5. The actor issues a 'Find Displays' command using the pull down menu at the top of the interface.

   *The interface for the camera starts searching for any 'displaying services' on the local network and finds those shown in the screenshot below:*

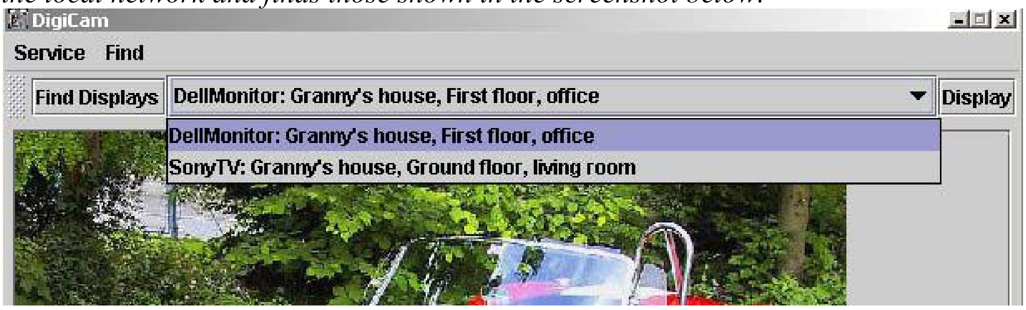



**Figure 7-20: Found displays.**

6. The actor selects SonyTV in the living room and clicks 'Display'.

   *The picture appears on the television screen.*

## Scenario: edit pictures

The sequence of activities for editing the pictures is almost identical to that given above with the exception that the photo editing service is provided not locally but by a company on the Internet (outside the local network). The actor knows there is a link to this service, (called 'PhotoPad,') in her 'Office CTI' place and uses CyberCMD to browse there (*via* HotSpot) and launch the interface for PhotoPad. The sequence is illustrated below:

```
:local>list
:local>go HotSpot
:local>hotspot>list
______ Services ______
Home
OfficeCTI

_______ Places _______
Home
OfficeCTI

:local>hotspot>go OfficeCTI
:local>hotspot>OfficeCTI>list
______ Services ______
CamService
DigitalClock
CashConverter
TextEditor
PhotoEditor

:local>hotspot>OfficeCTI>launch
PhotoEditor
```

**Figure 7-21: Screen output CyberCMD.**

*The following interface appears:*

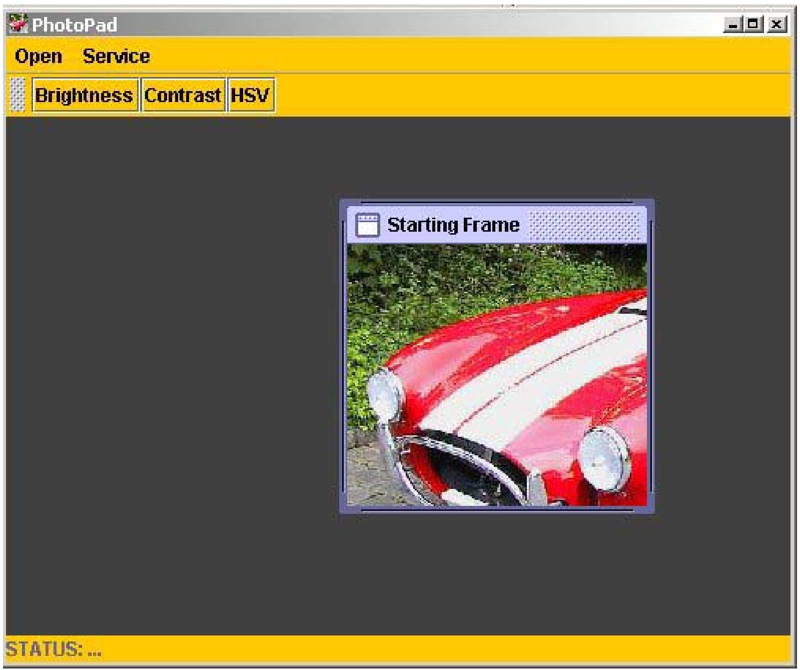



**Figure 7-22: The user interface for PhotoPad after opening content.**

*1.* Once the photo editor starts the actor selects 'open' from the menu.

   *The PhotoPad interface starts searching for local 'image provider' services and shows a dialog that displays them.*

2. The actor selects the digital camera from the list of devices from which she can then retrieve the images she wants to edit.
3. Having made her changes the actor selects 'Save' from the menu to save the changes in the image files stored on the camera.
4. She can then close the PhotoPad service by closing the interface.

The narratives above illustrate how a task from the scenario can be performed using the browser and the services developed for this project. It should be pointed out that all of the remaining tasks from the scenarios are accomplishable in the same manner.

## 7.5. Summary

This chapter introduced the implementation of the navigation layer (now called browser,) of the Services Platform specified in the previous chapter. It explained the inner workings of the browser and introduced the different user interfaces (text, 2D, 3D speech interfaces) that can be used with the browser. The services that are needed to implement the scenario were introduced. Some services were described in more detail to enable a their differences to be highlighted. The choice of organisation and metaphor used for the implementation was discussed and was followed by a description of the implementation of a scenario. The next chapter provides an evaluation of these elements.

# 8. Evaluation and interpretation

*In this chapter the results from evaluating the implementations introduced in the previous chapter will be presented. First the evaluation themselves will be discussed followed by the results and an interpretation of the results.*

## 8.1. Evaluation

As was seen in the methodology section (chapter 5) the results come not only from the *evaluation* of the prototype, but also from the *development* process itself. A process which, as well as building the prototype, also included the initial scenario design and literature review. Findings relevant to the hypotheses were recorded in writing during the development process and were analysed along with the data from the expert interviews and user testing. So taken together the sources of the results are:

- The scenario design.
- The literature review.
- The building of the prototype.
- Evaluation of prototype (by experts and users).

The DECIDE framework specified by Preece et al (Preece, Rogers, and Sharp, 2002) was chosen as the evaluation strategy to adopt. The questions to be answered by the evaluation were also selected, just as the paradigm and technique to be used. The evaluation is of a predictive nature and the techniques used were expert interviews and user testing. The latter consisted of questionnaires, open questions, and user-observation.

At the end of the research period each interview, questionnaire, and observation was analysed for answers, comments, and behaviour relevant to this thesis. The analysis was conducted manually as the data did not lend itself to formal analysis. During the analysis particular attention was given to finding similarities between subjects reactions. If more than one subject made the same mistake, gave the same comment, or asked the same question, this was interpreted as an indication of the importance of the issue for the subjects. The next section discusses the most important findings as they relate to the hypotheses.

### 8.1.1. Expert evaluation

Five experts were individually asked to give their responses to the prototype at different stages of the evolutionary design process. All experts worked as usability experts at a company or a university. The means used to gather their reactions took the form of informal discussions on usability topics related to this thesis. The topics discussed ranged from the use of sound in the interface, to the support of social navigation. Many valuable comments were made by these experts during the discussions and were used to guide further development. The prototype's final form was the topic of a somewhat lengthier and more focused interview with the experts. The interview was based upon a few open questions that served as a starting point for a frank discussion of the issues relevant to navigating services. The results from the expert evaluation were used to devise and structure the more formal subsequent user-evaluations. The base questions for the informal interviews were:

1. Do you think there is a fundamental difference between navigating functionality, in other words looking for behaviour, and navigating 'plain' information?
2. A difference between navigating information and navigating services might mean new opportunities for metaphors. Can you think of metaphors that are not being used for today's desktop computers?
3. The difference might also mean changes for the *ways* of navigation; e.g. social, spatial or semantic navigation. Where do you see new threats and opportunities?
4. Existing Web browsers requiring more and more plug-ins to display information or functionality other than Web pages, which indicates that they are not optimal for rendering services to the user. Do you agree?
5. The prototype might be implemented in many modes, meaning that the interface can be constructed in 2D, 3D, speech or something else. Do you think that the fact that services are being navigated influences this choice? And if so, how?
6. The evolution of Web pages showed that information about the page itself (metadata such as 'meta tags' for instance) was becoming increasingly important. Does navigating services as enabled by the prototype have effect on the metadata requirements from the services?

### 8.1.2. User tests

The next phase of the evaluation was to ask potential users how they felt about using the prototype, and observing them while they were using it. Because the prototype was designed to be usable by as wide an array of users as possible[240] I endeavoured to ensure a degree of diversity amongst the test subjects.

The subjects were first asked to use the prototype to perform several tasks taken from the scenarios[241]. In order to put the testers at their ease and, as far as possible, to

---

[240] See section 6.1.6.

[241] See the appendix (chapter 11) for the tasks involved.

prevent the results of the tests from being skewed by an 'experimental' setting the tests were set up in the following manner:

1. Desktop computers upon which the prototype was already running were used.
2. To put the subjects at their ease, and to bring the testing situation as close to 'real world' conditions as possible, the tests, where possible, were done in a familiar setting using the subject's own computers.
3. The subjects were encouraged to 'play' a little with the prototype to familiarise themselves with before they were asked to perform the tasks. This 'getting acquainted' period was also conducted under observation and I sat next to them to instruct, observe, and sometimes to ask them further questions. Both my observation of the users' behaviour and their answers to further questions were recorded in writing during the session.

Unfortunately I didn't manage to test the speech interface with users. The speech engine used by the interface requires a lot of training from each user, which would have taken too much valuable time. Besides that some preliminary tests showed that the interface worked too awkward to be really usable. The design of the interface did yield some valuable insights regarding the hypotheses, so no time or energy was lost in building it.

See section 11.2.1 in the appendix for a table that summarizes the most interesting observations. The appendix also contains a complete overview of all the (unique) observations and how often they occurred (section 11.3.1).

After finishing the test session the subjects were asked to fill out a questionnaire. The guidelines given by given by Preece, Rogers, et al (Preece, Rogers, and Sharp, 2002) and Chin, Diehl et al (Chin, Diehl, and Norman, 1988) were followed in designing the questionnaire. It consists of both open and closed questions. The closed questions had three possible answers:

1. agree,
2. disagree,
3. undecided,

and were divided between questions about the general use of the prototype, and questions on navigating services using the prototype. A summary of the results can be found in the appendix, section 11.2.2, where the questions are grouped by topic. A complete listing of all the answers are filled out in the questionnaire given in the appendix, section 11.3.2.

The questionnaire also contained some open questions that gave the subjects an opportunity to express their findings in greater detail and in a more personal way. It also had a ‘comments’ section, in which the subjects could fill out additional comments. A summary of the most interesting answers can be found in the appendix, section 11.2.3, and a complete listing of all the answers can be found in section 11.3.3.

## 8.2. Interpretation

In this section the main results from the different sources will be integrated and analysed for insight into the hypotheses. The interpretation of the results is divided by the topic of each of the three hypotheses.

### 8.2.1. Navigating documents vs. services

The first hypothesis formulated in chapter 5 is:

> *There are differences between how users navigate in search of information (documents) vs. how they navigate in search of functionality (services).*

The analyses yielded a number of insights concerning the difference between navigating information and navigating services.

**Becoming lost**

The scenarios made it clear that in some situations a considerable number of services could be available to users. Such a situation could well result in users becoming confused, leading to their becoming lost and unable to find the services which they seek. The literature study showed that unlike navigation of the Services Sphere there is already a substantial theoretical background on this topic as it relates navigating Information Space goes (e.g. (Theng, Thimbleby, and Jones, 1996) and see also chapter 4). This is because navigating large amounts of services is not as yet a practical problem[242] although it may become so as the number of available services expands. Whether becoming lost will be a problem could be ascertained by using the prototype to implement and test a future scenario.

The goal of the study was to highlight relevant issues and possible solutions to them. None of the users who took part in the testing phase had any prior experience either with using a service-oriented architecture, or with navigating distributed services (the Services Sphere). While all of the test subjects had experience in using applications on computers, and were effective in navigating that environment, none of them had previously navigated a collection of services distributed across the Internet. It should also be said that differences among test subjects lay in their familiarity with different interface modes, or metaphors.

---

[242] See section 4.5 for the state of the art in service-oriented systems.

The user tests indicated that becoming lost is still a problem for users navigating services, even with the relatively small number of services of the test set-up. Although not all subjects had problems during the tests using the text-based interface some definitely did. Those who experienced problems were hesitant and sometimes had to ask the supervisor how to proceed. Such users also mentioned the feeling of being lost in the interview or questionnaire. This situation improved with the 2D and 3D interfaces but some users still felt lost at some point.

One noteworthy finding was that those users with the most computer experience had the least problems of becoming lost, particularly when they had had previous experience of using a text-based interface. All of the users had used 2D interfaces before, and the test showed that, on average, the problems of becoming lost while using such interfaces were somewhat lessened. This confirms that user understanding of the Services Sphere is influenced by the interface mode in the same manner as in Information Space. A conclusion supported by, for example, Card, Mackinlay, et al's research into information visualisation (Card, Mackinlay, and Shneiderman, 1999)). It would seem that using an interface mode with which users are familiar is a factor in helping them navigate the Services Sphere. It should however be said that interface *mode* of itself is not a sufficient aid, but needs to be taken into account in conjunction with the *metaphors* used and the *organisation* of the services. Both of these other variables are identified by the separate layers of the hierarchy of navigation seen in section 3.3. It will be recalled that that the layers are often closely connected and the effect, the freedom of action if you will, of the upper layers is limited by those lower in the hierarchy. The organisation layer's effect is felt mainly in the metaphors it affords as the metaphor used directly affects user understanding of the Services Sphere. It should also be said that the users reacted positively to the geographical metaphor saying that it had helped them to understand the Services Sphere – a point to which we will return in the next section.

It seems that, as with navigating Information Space, experience in using an interface mode, coupled with an appropriate and readily understood metaphor, can decrease the chances of users becoming lost as they attempt to navigate the Services Sphere. This raises the question to which extent each layer can help prevent the user from becoming lost. Are there more such opportunities than with Information Space? The answer to this question comes from the organisation layer. As was seen during the discussion on the navigation of Information Spaces[243], superimposing a structure upon something that is inherently unstructured is not very effective (Dieberger, 1995). It is effective however to superimpose such a structure upon things that do have an inherent structure — such as services with ties to real physical places. As was highlighted by the scenarios many services do indeed have such ties, either because real physical objects provide them, or because they deal with topics that are closely related to

[243] See section 3.3.

physical places[244]. The implication of this is that the Services Sphere can be said, at least in part, to be significantly inherently structured. The advantage afforded by this is that, although they are free to do so, developers do not *have to* attempt to provide a metaphor that makes an effort to provide structure. A metaphor that communicates an inherent structure, such as a geographical metaphor for those services linked to physical places, is an appropriate and 'natural fit.' It can therefore be said that the inherent structure of the Services Sphere makes it easier for a combination of metaphor and interface mode to reduce or even prevent user confusion.

The value of the Services Sphere having an inherent structure was indicated by the fact that some users did not feel lost at all, even though they were navigating the Services Sphere for the first time. They had a clear conceptual model of the system, always knew where they were, and felt in control all of the time. As one user put it:

> *"I actually had the feeling of physically walking around in the places".*

Their behaviour was as though they understood the system without having to consciously think about it, in short, it came naturally to them.

Users who had used text-based interfaces before initially had an advantage. However, most of the users without such experience caught up surprisingly quickly after using the text-based interface for a short time. As one of them said they needed to, "get used to the system and how it works", that being accomplished they proved equally effective as more experienced users in using the system. This shows first that unfamiliarity with an interface mode can slow down the learning process for navigating services, and second that none of the users had fundamental problems learning to navigate the Services Sphere. While some users experienced problems with the interface mode, in particular the text-based interface, this is largely attributable to the fact that most users had little or no experience with such interfaces.

Differing levels of experience cannot alone account for the difference in becoming lost between experienced and less experienced users, as even experienced users got lost at some point during the test. It seems that becoming lost is a problem that arises when navigating the Services Sphere as well as when navigating Information Space. The similarities between the two situations are obvious; the user tries to navigate a virtual environment using a computer. This requires a different cognitive effort than navigating a real environment would. One subject highlighted an interesting point with regards to this when he said:

> *"Mentally I was where I was physically, right in front of a PC".*

---

[244] For instance a service for finding and reserving a tourist accommodation

Such a reaction illustrates a difficulty associated with navigating virtual environments; users will have a dual mindset, one for the *virtual* world, and one for the *actual* world. It is possible that with media where the user cannot escape the virtual world[245] this dual mindset could disappear, however such interfaces were not tested with the prototype.

One way of diminishing the chances of users becoming lost may possibly be found in the guidelines for constructing easily navigable cities developed by Kevin Lynch (Lynch, 1960). As Dieberger (Dieberger, 1994) points out, many system designers hoped to minimise, or even eliminate navigational difficulties by applying Lynch's guidelines for Information Space construction[246]. Their attempts met with varying levels of success, and whether his guidelines can be successfully applied to the Services Sphere could be a useful topic of further study.

Building the prototypes showed that there are different ways the interface can prevent the user from becoming lost. Which user-support can be provide largely depends on the interface mode. A 2D and a 3D interface for instance offer the opportunity to display a map of the history of visited places, while this is impossible with a text- and speech interfaces. The difference became clear when building the prototypes, where for instance the text interface had a history list but the 3D interface didn't. This shows that to prevent users from becoming lost designers have two options. Firstly they can construct an easy navigable Services Sphere, as discussed in the previous paragraph. Secondly users can be supported by constructing user-friendly tools to navigate the Services Sphere. Which approach is taken depends on the question which variable, the environment or the tool, the designer has sufficient control over.

## Finding services by their connection with real physical places

During the tests the users were asked to complete a task such as 'launch the text editor' without being given any indication as to where this service might reside. In a few cases this led to the user at first going to the wrong location, and the proper location immediately afterwards, however most of the users found the correct place for the service without such delay. In navigating an unstructured Information Space like the Web an unspecified task such as 'get the weather information for Tuscany' could result in two types of behaviour. If the users were to know where to find the information they could type in a URL in the address bar of the Web browser, use a bookmark, or go to another page from where they know they can navigate to the required page. If, on the other hand, the users do *not* know where to find the information they might use a search strategy, be it using a search engine, using a directory, asking somebody who might know the answer, or something comparable.

---

[245] As is for instance the case when wearing a virtual reality helmet.

[246] See section 3.1

When it comes to the Services Sphere, its inherent structure makes a geographical metaphor possible rendering the need for searching less likely. Even with unknown services users will often have an idea where they are most likely to be found. This was for instance the case with the instruction to change the temperature of the refrigerator during the user tests. All users without hesitation navigated to the kitchen place to indeed find the requisite service. They commented that the kitchen was the most obvious place for a refrigerator. This approach wasn't always without problems though, when asked to check the time all users started searching for a clock, but some expected a clock in the living room, while others expected one in the kitchen. It showed that personal experience in navigating the physical world makes a difference.

When building the prototype services it turned out that for some services the designers have control over how much they stress the connection between the service and physical reality. The name of the service can for example reflect the hardware that provides the service. A service for controlling the lights might be called 'LightSwitch Service'. This was the approach taken when building the prototype services, and breaking with that approach sometimes caused confusion during the user tests. The service for programming the video recorder was for instance called 'VCRX', but most of the users didn't realise that it had something to do with a video recorder.

## Launch service from a browser or from another service

When it comes to navigating the Web users, most of the time, navigate from *within* one Web page to another. This is because the link to the destination page generally is placed inside a Web page. There are exceptions to this rule such as the use of favourites (bookmarks) or the 'back' and 'forward' buttons in the Web browser. The situation is similar for services as the link to other services can be presented either by the browser, or by the interface of the service itself. Thereby allowing users to launch a new service interface both from within the service browser, or from inside another service. The 'Travelling Pictures' scenario[247] showed an example of such activity where the actor was able to open a printer service from within an image editing service. Also the design of the prototype services often begged the question whether to put a link to another service inside the interface itself or leave that to the service browser. What turned out to be the best solution differed per service and depended on things such as the interface mode, the service being linked to, and the number of services being linked to.

The paragraph above calls attention to a difference between navigating an Information Space such as the Web and navigating the Services Sphere. For the most part the Web is navigated by going from the *content* of one document to the content of another. The relative homogeneity of Web documents makes this possible because, in general, differences between documents are not sufficiently large to cause confusion. However

[247] See section 2.3.

this cannot be said to be the case with services interfaces. Services interfaces typically have higher user interaction requirements making them not only less homogeneous, but also more likely to disrupt a previously seamless user experience. In other words it could well be harder for users to navigate the Services Sphere because each service interface encountered might not only look completely different, but may also have a proprietary way of linking to other services. Two of the services developed for the user tests offered links to other services in different ways. Where one offered a link in a menu the other had a button in the main interface that linked to another service. This can confuse users, as was made clear by the user test during which most users experienced difficulties finding (other) services from within a service interface, a difficulty they did not have using the browser.

For other kinds of Information Spaces there can also be ways to navigate between documents without going 'through' each document. An example from the Web could be a map of a company's Web site (often called a 'site map') or the previously mentioned bookmarks or back and forward buttons of a Web browser. Besides that many studies in information visualisation (Card, Mackinlay, and Shneiderman, 1999) can serve as examples. LifeLines, TimeScape and LifeLines as discussed in section 3.5 are examples of visualising information that provide a method for navigating between documents without going 'through' the documents.

## Start with service or content

Another difference between navigating Information Space and navigating the Services Sphere is that the latter allows users to either start with a service and find content, or start with content and find an appropriate service. This difference was seen in the analysis of the 'Travelling Pictures' scenario[248] when the subject had the choice of both starting with an image editor service and opening an image to work on, or of starting with an image and finding a service that could be used to edit images. - A situation that does not occur when users are only navigating documents, as documents tend not to act upon each other. When constructing some of the prototype services the same decisions had to be made.

One of the tasks from the user tests required the test subjects to print an image from a digital camera. This could be done either by starting with the image and finding a printer, or by starting with the printer and then locating appropriate image content to print. It transpired that most users accomplished this task by starting with the image and then locating a printer that could deal with it. It should be said that while most of the users accomplished the task in this way that, in general, they commented that they had no preferences in how to accomplish the task, that both ways were equally viable. This raises the question of why most of the test subjects chose to start with the image. It could be that they chose to do so because they had already seen the images while

[248]See section 2.3.1.

accomplishing another task and therefore knew where to find them. Further research is needed for a definite answer.

## Entering data => Privacy. Personalisation. UI requirements

That services have higher physical interaction requirements than documents was first hypothesised by the scenarios and subsequently confirmed by both the development of the prototype[249], and the user tests. Services often require input from users, whether it be, pressing a button, entering a password, selecting an option, opening a file, playing a sound, or some other action is irrelevant, all such activity consists of more interaction than simply reading a *document*. This higher level of user interaction affects the services' user interface, which has to accommodate the higher requirements. Higher interaction requirements also have an impact upon such matters as security, personalisation, privacy, and preferences. User data entered into a service may require to be rendered inaccessible to others in order to protect user privacy, or for reasons of commercial confidentiality. Such privacy and security requirements are relevant to many types of data such as passwords, medical data, financial data, personal documents such as diaries, and so on. Thus if a user were to use a text editing service to record entries in their diary[250] the service might offer to store the diary securely on the server side. This requires different, and thus potentially more, security to be in place than with merely reading a document. The diary has to be first opened, and then stored again after the user has finished - both which are operations that require authentication, secure connections, and trust between the service provider and the service consumer.

The importance of such issues as authentication, secure connections, and trust, was emphasised by the development of the prototype. There were many security issues to be dealt with both for the browser and the services in general. Moreover the differing nature of the services meant that each service had requirements peculiar to it. Sometimes the only user input required was to click a button to get a temperature reading from the refrigerator, but at others the input could be very personal and sensitive data such as the mentioned diary entries. During the user tests some users asked how certain services (for instance turning up the heating in the house) could be protected from unauthorised users. Such questions reminded me of, and re-emphasised, the importance of such issues. It is likely that unauthorised use of a service could potentially do more harm than the unauthorised reading of a document, and the tests showed that users seem to be well aware of, and concerned about, this aspect of service provisioning.

---

[249] This has been discussed with the requirements for the prototype in section 6.1.

[250] A task in one of the scenarios.

The fact that users provide services with some input means that security and personalisation are more important for services than for (most) documents. The centrality of these issues affects users' navigational behaviour *between* services. Notwithstanding the fact that issues such as security are a prime importance to many service providers, users may well hesitate to launch a service new to them because of security concerns. In other circumstances they might decide to trade security for the convenience of avoiding having to login to some services each time they launch them. For example by using a service that provides automatic authentication to other services. That such issues (and tradeoffs,) are of top priority to users was made clear by questions asked by some of the test subjects before launching certain services. Plainly such issues need to be addressed by designers, developers, and providers of systems that navigate the Services Sphere from the moment they start.

## Access from varying locations and circumstances

As was initially pointed out by the scenarios, users may want access to services from a greater variety of circumstances than they do with 'plain information', a point further stressed by the design and construction of the prototypes.

First, the service browser being used has to be usable in a greater variety of locations and circumstances. It should, for instance, support the use of voice-activated controls for those situations where the users do not have their hands free but want to use a service nevertheless. That users can access services from such a diversity of circumstances and locations using a variety of methods will plainly have an effect upon how they navigate.

The other effect is that it is likely that the range of devices used for accessing the services is wider than that for solely navigating information. Users could either have different devices available that could accomplish a task, or they might *prefer* to use particular devices in particular situations. In one of the scenarios the actor was driving a car while simultaneously accessing services. Plainly the devices she could use in that situation are quite different from those that she could have used were she sitting behind her desk at work. This is not to say that users will not access Information Space from such situations, they probably will, it is merely to point out that the addition of services makes it more likely that they would like access from a wider range of situations and therefore will require a wider variety of interfaces and devices capable of supporting them. The addition of services to Information Spaces therefore makes it more likely that users will demand not only increased availability of access, but also a range of devices that grants them such access in widely differing circumstances.

## Response time expectations

The way in which the test subjects behaved during the tests and their responses in the questionnaires indicates that users still very much think in terms of navigating

Information Space instead of the Services Sphere. They expect identical levels of the responsiveness when browsing the Services Sphere, as those to which they have become accustomed while browsing the Web. Such expectations cannot be met for the following reasons:

1. In most cases services are larger in size than plain documents, leading to an increase in download time.
2. Displaying the services will take a little longer because rendering a full-blown interface takes more time to process than displaying plain text.
3. In some cases a service is an aggregation of other services, which means that all sub-services have to be retrieved before the aggregated service can be rendered to the user.

The combination of these three factors make the experience of navigating the Services Sphere somewhat slower than users are used to with navigating the Web. Most users had to get used to this fact according to their behaviour and responses. When for instance launching a service interface they expected it to appear just as fast as the interface of an application that is installed locally on the computer, which was never the case. Feedback on how long it could take to retrieve the service proofed useful here, as found out during the evolutionary design process; users patiently waited for the maximum retrieval time. After a while the test subjects grew accustomed to the fact navigating the Services Sphere was slower than navigating the Web and became more patient.

Some of the users indicated that it took them time and a conscious mental effort to become accustomed to the fact that they were navigating services, rather than information. They needed to remind themselves of the differences involved and that this slowed the 'acclimatisation process.' This leads me to conclude that the differences between the two navigation types were not communicated clearly enough by the prototype.

## Overall findings

Other than the difficulties outlined in the findings thus far, it could be said that navigating the Services Sphere did not introduce any fundamental barriers. With few exceptions it transpired that:

1. Most users found it rather easy to learn the new way of navigation.
2. Most users felt in control while using the system.
3. And, even though in some cases it took a few attempts at using the system, most users had a clear conceptual model of it.

### 8.2.2. Geographical metaphor

The second hypothesis formulated in chapter 5 is:

> *A geographical metaphor is more useful for services with an 'actual geographical location' than it is for documents.*

The literature study[251] showed that attempts to use a geographical metaphor for Information Spaces were not very successful, giving rise to the question of whether the same metaphor would be usable for navigating services. The scenarios implied that a geographical metaphor *could* be useful for the Services Sphere, but an indication of the final verdict on this topic, comes from the user responses collected during the user tests.

**Useful metaphor**

The hypothesis deals with whether a geographical metaphor would be of more use for those distributed services that have links to an actual geographical location. This in turn raises the question of "useful for whom?" The ultimate answer to this should be "the users", as user requirements regarding navigating the Services Sphere are the focus of this research. When one considers the findings from the scenario design, the literature study and the user tests, the hypothesis turns out to be valid. The paragraphs below give a somewhat more detailed answer.

It will be recalled that the scenarios showed that many services are in some way connected to a real physical place, that is they have an actual geographical location. Examples from the 'The Trip' scenario are services to control the heating, the light, and the refrigerator in the cottage, all of which are services that are connected to a physical object, providing them with an actual geographical location. (The tables in section 2.3.1 list all of the services from the scenario.) The scenarios showed that even a small scenario contains many services with an actual geographical location. This does not necessarily mean there will be more services associated with a particular geographical location than those without such an association[252], merely that there are likely a number of them. Moreover a rise in the number of autonomous devices capable of providing a service[253] means that this number will increase.

For a Services Spheres containing services with an actual geographical association, it turned out to be natural to use a geographical metaphor that maps to the real

---

[251] See chapter 3.

[252] The instant messaging service mentioned in the scenario is just one example.

[253] A recent initiative from IBM called 'autonomic computing' focuses upon precisely this subject. More information can be found at this location: http://researchweb.watson.ibm.com/autonomic/

geography. Such a mapping allows users to use their knowledge of navigating the real environment in the virtual environment. As was explained earlier[254] as the number of services with an actual geographical association increases user understanding of the Services Sphere is likely to be enhanced by a clearly conveyed geographical metaphor. The advantages of such metaphors when clearly expressed were shown by those user tests in which users could easily find their way around without needing guidance. An example of such a situation occurred when the test subjects were asked to check the time. Each user did the following:

1. They asked themselves where they would normally find a clock (for some this was the kitchen, for others the office).
2. They navigated to that place.
3. And (in most cases) successfully found the clock.

Problems arise when services either do not have an actual geographical location, or the connection is not that obvious, as is the case for example with a text editing service. That such a difficulty could arise was first intimated at during the development of the prototype[255] where it was hard to implement a geographical metaphor for such services. That this was a difficulty was confirmed by the user tests in which the test subjects experienced more difficulties in finding such services. To continue using the text editing service as an example, many users were uncertain as to where such a service might be found. They resolved that problem in the light of navigation that they had already done. During previous 'excursions' they had seen the office room, and, most of the test subjects remembered correctly that that was the most likely location for the service. What this shows is that the usefulness of a geographical metaphor is to a considerable extent dependent upon the service to which the metaphor is being applied. Some services might not be provided by a physical object but still have an obvious correlation with one, thus a recipe service could have an association with a kitchen. By way of contrast services such as an instant messaging service have no such immediately obvious linkage. — The more obvious the connection, the more useful a geographical metaphor.

As the hierarchy of navigation shows[256], the fact that many services have an actual geographical location does not mean a geographical metaphor *must* be used. It *is* a natural fit in such circumstances, but such a fit does not of itself mean that other metaphors cannot be equally appropriate. The usefulness and appropriateness of other metaphors *vis a vis* the geographical one as it relates to Services Sphere navigation might usefully be a topic of further research.

---

[254] When discussing 'becoming lost'.

[255] See section 7.3.

[256] See section 3.3.

**Relation between the physical and the virtual geography**

The other question raised by the hypothesis regards the *nature of the relation* between the physical and the virtual geography. While a physical geography denotes where *objects* (whether or not associated with services) reside in the real world, a virtual geography denotes where the *service* resides in the *virtual* world. The issue now arises as to how these two geographies relate to each other for those services that have an actual geographical location, meaning they have a link with real physical objects (be it a room, a VCR, or a doorbell).

As the discussion above showed the relationship between a service and the real world is very much dependent upon the following two factors:

1. Whether a real physical object is serving it.
2. Whether it has a strong association with a physical object.

The implementation of the services and the prototype made it clear that it can be hard to make a clear distinction between those services that do have an actual geographical location and those that do not. Rather than being comprised of clear alternatives it is more like a dynamic sliding scale. When a physical object provides the service the connection is both clear and immutable, such as a video recorder providing a service to program itself. For other services the connection is less clear, such as with the text editing service mentioned above. For some services the connection is not at all clear, a calculator service might be one example. The scale's dynamism comes from those *situations* in which a service might have a more obvious connection to an actual geographical location than in others. A cookery book service might be associated with a both a library and a kitchen, a text editing service might be appropriate only to an 'office room,' while a loan repayment calculation service might be found either in an 'office' or in a virtual city's 'financial services district.' — It depends on the situation from which the user is looking for such 'ambiguously located' services.

If all physical objects had a virtual presence the real world could be completely mapped to the virtual world. Taken to extremes this could, at least in theory, mean that the difference between the two would become unnoticeable. However as the scenarios and implementation of the prototype pointed out, by no means all objects are equipped with virtual counterparts, a fact which would tend to make such a mapping rather difficult to implement. Furthermore, as was mentioned earlier, the extent to which services in the Services Sphere have a physical geography is to some extent situational. Navigating the virtual world of services will therefore never be exactly like navigating the real world.

### 8.2.3. Metadata

Both the scenarios and the literature review showed that it is likely that different user interfaces for navigating the Services Sphere will entail a diversity of metadata requirements from services. This lead to the formulation of the third and final hypothesis in chapter 5:

> *Different metadata is required from services than from documents to implement the different navigational user interfaces.*

Whether this is a valid hypothesis will mainly follow from the development of the prototype and the services necessary to implement the scenarios.

As discussed in previous chapters the user interface is a tight combination of the organisation-, metaphor- and modal layers of the navigational[257] hierarchy. Each of the layers has a specific metadata requirements making it is useful to divide the discussion accordingly. It will be followed by a discussion about the general issues.

## Organisation Layer

Using a variety of organisational methods obliges services to provide different kinds of metadata. Thus, in a time-based organisation where services are organised according to the time they were last used the service would have to have metadata regarding when it was last used (either in general or by a particular user). Regarding a location-based organisation metadata is required about the geographical location of the service. There are two possible sources for data about the location of the service. Firstly the data might come from the geographical location of the device that offers the service. Secondly the data might come from the semantic connection a service has with a geographical location; a 'recipe service' might be associated with the kitchen, while a 'game service' might be associated with a leisure room. For the types of organisation that were part of the study the following metadata requirements were identified:

| Organisation | Metadata documents | Metadata services |
|---|---|---|
| **Time** | - Last time viewed<br>- Last time modified | - Last time used<br>- Last time changed preferences |
| **Location** | - Semantic connection with a location | - Location of the device providing the service[258]<br>OR<br>- Semantic connection with a location |

[257] See section 3.3.

[258] If the service has a clear connection with a device, e.g. is provided by it.

Other types of organisation could be an activity-based organisation, which requires the service to provide metadata about the type of activity with which it is associated. Yet another way of organising services is to divide them into activities related to work and to leisure time. The category to which a service belongs could be specified by the metadata[259].

Unfortunately, there is no *simple* answer as to what additional metadata is needed from services to facilitate different ways of organisation. The main conclusion that can be drawn is that different organisational schemes have different requirements.

## Metaphor Layer

All metaphors have unique metadata requirements. The geographical metaphor used in the study for instance required the service to provide any available geographical information. In the 'Travelling Images' scenario Alana needed to know which of the available displays she could use to give a slideshow. Information about the location of the display was provided as metadata with the display services and could be used for a geographical metaphor. A geographical metaphor requires the following metadata:

| Metaphor | Metadata documents | Metadata services |
|---|---|---|
| **Geography** | Geographical location | Geographical location |

The geographical metaphor requires information where to place the services in the virtual environment. Similar to a location based organisation this information is likely to come from two possible sources. Firstly, if the service has a clear connection with a device the information can come from the geographical location of the device, as was the case in the above-mentioned scenario. Secondly the information can come from the semantic connection a service has with a geographical location.

As with the organisation of the services there are no *simple* answers as to what metadata is needed in order to accommodate each metaphor used. It depends upon the metaphors themselves, different metaphors require different metadata; a desktop metaphor for instance requires information where the icon for the service should go on the desktop, whereas a workflow metaphor[260] requires information about when the

[259] It should however be said that some activities, for example Web page authoring, might be categorised as work by one class of users and as a leisure activity by others, stressing the fact that metadata depends on more than just the service. Such dualism could be handled by interaction between metadata and preferences specified by the users.

[260] A metaphor that orders the services according to the time the user worked for the last time on the document.

service was used for the last time. The requirements will have to be identified separately.

## Mode Layer

The mode layer is situated on top of the metaphor layer in the hierarchy of navigation, and influences the metadata requirements as well. The interface modes given by way of example in section 3.7 are text, 2D, 3D, speech, and haptic. Each of these interface modes that are usable by a navigational interface (or 'browser') requires different metadata from the services as well. Thus when developing the text-based interface a clear description of the service was potentially more important than with, say, a 3D interface because the latter might also communicate information visually. This was borne out by the user tests of the text interface during which the test subjects relied heavily upon the descriptions provided. Indeed one test subject commented that it was sometimes hard to discern a service's purpose based solely upon its name and that the description was useful in that respect.

It also it became clear while developing the 2 and 3D interfaces that 2D/3D icons were needed for each service, this was not the case with the text-based interface. The icons in question were made part of the metadata for a service.

The table below summarizes the metadata requirements that were identified during the study. I won't discuss each type of metadata in detail since most of them speak for themselves.

| Mode | Metadata documents | Metadata services |
|---|---|---|
| **Text** | | |
| **2D** | | - Service specific 2D icon |
| **3D** | - Coordinates where to position 2D icon | - Service specific 3D icon<br>- Coordinates where to position 3D icon |
| **Speech** | - Sound icons of all metadata | - Sound icons of all metadata |
| **All modes** | - Name<br>- Document type<br>- Access rights<br>- Owner | - Name<br>- Description<br>- Access rights<br>- Security information<br>- Service provider<br>- User preferences<br>- System requirements |

The tight combination of the three layers of the hierarchy of navigation means that a decision made at one level influences the other levels, something that is also true for metadata requirements. If the developer for example were to choose an activity-based organisation, that decision would limit the number of useful metaphors, and would have an influence upon the metadata requirements.

## General issues

An important conclusion that can be drawn from the implementation of the different navigational interfaces is that services have higher metadata requirements than documents. This is mainly due to the fact that a service is more complex than a document. A service requires more interaction with the user, can have more than one user interface. Furthermore, it can cooperate with other services, and can behave dynamically, as well as potentially having high security and privacy requirements. Put another way, there is more to be known about a service, and therefore it has higher metadata requirements.

As the examples used thus far have shown, metadata cannot only take many forms, but can also be 'user' specific. This is particularly true for service metadata. As was explained in section 4.1 (and confirmed during the implementation of the prototype) it is likely that many services will cooperate with other services *without human intervention*. The image editor service given is an example of such a service. For successful cooperation enough metadata needs to be associated with a service to enable other services to determine whether it offers functionality they require. The fact that metadata can be used by humans, by 'traditional' software, and by services, can result in widely differing forms of metadata. — A description of a service is entirely different from an encrypted piece of binary information used for user authentication. Indeed if one were to carry this to an extreme a handwritten manual could be regarded as metadata. As was seen above it is not specified in advance who (or what,) organises services, either humans or software can do it (for instance a browser that reads services' metadata and organises them accordingly). Both situations require different types of metaphors. The software needs a machine-readable format, while the human needs a human-readable format.

One question raised during the development of the prototype and services is how to associate metadata with services. There are two aspects to this question. The first of which deals with the question where the metadata is to be stored. Multiple options were identified during the development process. It could be stored with the user, the service, in a third location, or some combination of these. The browser used for navigating the Services Sphere could, for instance, track the services a user has used, and store specific metadata for each of them. The metadata could also be stored with the service itself, which is a natural option for service specific metadata, but rather less so for user specific metadata. Another alternative is to store the metadata in some remote location accessible by both the user and the service. An example of such an arrangement is a UDDI registry of Web Services described in section 4.5.2. Such a service comprises of nothing more than a large collection of *descriptions*, or metadata, of Web Services. A final choice could be to use some combination of these alternatives. Storing user specific metadata with users, and general service metadata with services would be such a hybrid. Each approach has advantages and which method is best is situation dependent.

The second issue raised is how, and indeed to what extent, metadata can be changed once it has been associated with a service. Metadata can be subject to changes due to such factors as: usage, time, and preferences to give but three examples. Where metadata resides, largely determines the means by which it can be changed. If it is stored with the service it is the task of either a human service administrator or an administration service to change it, while if it is stored with the users it is the task either of the users, or an agent acting on their behalf.

While standards for associating and updating metadata for documents are devised by projects such as the Semantic Web project[261]; the same does not apply to services. Given that services have rather different metadata requirements than documents it is impossible to directly translate the standards from Information Space to the Services Sphere. It is up to subsequent research to define the new standards.

## 8.3. Summary

This chapter summarised the results from the research and discussed the insight it gave into the validity of the three hypotheses given in chapter 5. First the evaluation of the prototype and its accompanying services was discussed, followed by an interpretation of the results. The results came from a variety of sources; the scenario design, the literature review, the building of the prototype, and the user tests. The first research question focussed on the differences between how users navigate Information Space and how they navigate the Services Sphere. Some similarities and many differences were identified. The usefulness of a geographical metaphor for services with an actual geographical location was discussed next. While the final topic discussed was services' metadata requirements as they related to different navigational interfaces. The discussion was guided with reference to the hierarchy of navigation. The next chapter will bring the results to a conclusion and examine future research possibilities.

[261] See section 3.8.

# 9. Conclusions

*There are fundamental differences between navigating services and navigating documents. The research indicated that the differences are due to the different nature of services, which results in different uses. Furthermore, the fact that services are often provided by a particular physical object offers new opportunities for organising the services. Another consequence of the latter is that a geographical metaphor is probably more fitting for services than for documents. Finally there is a difference between the metadata requirements for documents and services. They are closely related to the organisation, the metaphor, and the mode used. As with the topics of the three hypotheses the conclusions are divided into three parts. This discussion will be followed by an overview of possible future research work, after which some concluding remarks will finish this thesis.*

## 9.1. Main Conclusions

The goal of this thesis is to outline the issues (and non-issues) in navigation when moving from local documents to distributed services. The most important conclusion that can be derived from the research is that there are important differences between navigating local documents and navigating distributed services. What follows is an indication of what those differences are and what questions need to be asked. Because of the qualitative nature of the research the results should be interpreted not as a "*ne plus ultra*" but indications that can act as a stimulus to further research.

### 9.1.1. Hypothesis 1: Navigating services vs. navigating documents

As said, the research supports the first hypothesis; there are differences between how users navigate Information Space and how they navigate distributed services. These differences are likely to be accounted for by fundamental differences between documents and services.

The building of the Services Sphere suggested that it has an inherent structure due to the ties between many of its services and physical entities. As the number services with such a connection increased the structure became clearer. A clearer structure in turn made it easier both to organise services and to find a useful metaphor. The user-tests in turn showed that the latter enhanced user understanding of the Services Sphere. That the structure inherent to the Services Sphere is not under control of service browser designers requires them to help users by, for example, accenting the linkage between services and their geographical locations where such linkages are present.

The Service Sphere's inherent structure can ease navigation, the research indicated. The configuration of the physical objects can serve as a blueprint for the organisation and metaphor of the services in the Services Sphere. Users' familiarity with the physical environment can then be used to navigate the Services Sphere. By correlating the location of physical objects to locations in the virtual world, designers can enhance users' navigational experience by exploiting their familiarity with the physical world.

The scenario design highlighted the issue that services can be accessed from either a browser or other services. A service browser is one way to access a service, but it turned out that some services could eliminate the need to use a browser by making direct access to other services available. This means that a user can navigate between services either by using a browser that provides an overview of services, or by going directly to a service from *within* another service. The subjects of the user study indicated to have no preference, but the scope of the study doesn't allow to unambiguously answer the question whether users prefer to find services by using a browser or by using other services. The user test *did* show that it depends on the interface how easy it is for users to access other services. A lack of a standardized way to provide links to other services and complex service interfaces will make it harder to provide an unspoiled user experience. In the test set-up a few services linked to other services, but each in its own unique way, which led to confusion of the subjects.

In many cases services will require more physical interaction than just reading, as is the case with documents. The reason is that services are used for accomplishing a task rather than providing information. This means that the nature of the interaction is different between services and documents. The effect might be that the more demanding physical interaction requires services to have more complex user interfaces. The prototype services as developed for the user-test showed that the amount of interaction required can vary from as little as requiring a button to be clicked to save some text, to the complex sequences of activity required for displaying the images from a digital camera on a nearby television. The high interaction requirements resulted in the need for complex user interfaces, and although this research is about navigating *between services* (as opposed to *within services*), this is an important factor because links between services can be provided from within the interface of a service, as discussed above.

The implemented scenario contains a situation in which the subject wants to perform an operation upon a picture. This highlighted the question whether the subject should start with the picture and find the appropriate service, *vice versa*, or have a choice. In other words; with services users might have the option to start with service and find content or vice versa. It was decided to test both strategies to get an indication of the user preferences. Due to that small test group a significant general preference could not identified but it turned out that the test-subjects' preferences depend on the task and their experience with similar systems. For the task of editing a picture for instance, some users said to start with the editing service because that resembles the way they are used to when editing content. They use their experience with documents and applications on standalone desktop computers to combine network distributed services and content. Others started with the picture and try to find the appropriate service because that was the most logical order of actions for them. Both strategies involve different navigational approaches that should be facilitated by a service browser.

The indication that services require more physical interaction brings with it a need for higher levels of security, personalisation, and interface standards. Both services and the browser ought to meet these requirements. When developing the browser many choices had to be made how to handle security issues such as running downloaded code, and providing passwords and preferences to services. Many of the developed services also required security choices. Some services need to be prevented from unauthorized use, a point made clear by the prototype services that can be used to control devices. Personalisation is more important with services than with documents because higher interaction requirements mean a bigger chance of differences in user preferences. Higher interaction requirements can only be met when they are supported by the interface. This means higher requirements for the way the interfaces are built.

The scenarios made clear that users will likely want to access services in different circumstances than those that apply to documents. Both the services themselves and the service browser would therefore have to accommodate use from a variety of circumstances. This could mean providing user interfaces appropriate to a variety of conditions. Examples of such interfaces might include an interface that is designed to be used during the day, and another appropriate for nighttime conditions.

Some of the subjects of the user-tests initially had the same expectations regarding the task of navigation for the Services Sphere as they have for the Web. This became especially obvious with the slower response times when navigating the Services Sphere. Mentioned users were initially surprised by the time they had to wait for a service to appear on their screen after launching it. The development of the services and the services platform indicated that the differences between Information Spaces such as the Web and the Services Sphere are however, considerable; the organisation might well be different, larger files give rise to longer response times, launching a service might entail a certain amount waiting as the service is downloaded and initialised, to name but a few. The two are likely not the same, and service browsers should communicate this clearly to the user.

While navigating documents and navigating services have differences they also possess similarities. One important parallel is the indication that users can become lost while navigating both Information Space and the Services Sphere, where lost means not being able to find one's way. Even though the scope of this thesis did not allow putting all possible combinations of service organisation, metaphor and interface mode to the test, some of the users became lost at some point while using one combination or the other.

The user-tests suggested that the problems of becoming lost diminished when an interface mode is used with which users feel more comfortable. The problems surfaced especially with the text-based interface, an interface mode that most of the subjects didn't really like. A subsequent test with the more familiar 2D interface showed a sharp decline in problems of becoming lost, although they didn't completely disappear. Furthermore, comments from the subjects indicated that the clear geographical metaphor helped them understand the Services Sphere, preventing them from becoming lost. This lead one to argue that using familiar types of organisation, metaphors, and interface modes, may help smooth the learning curve for users.

### 9.1.2. Hypothesis 2: Geographical metaphor

The hypothesis that a geographical metaphor is more useful for services with an 'actual geographical location' than it is for documents turned out to be valid within the scope of this research. The connection between services and unique physical objects turned out to be fertile soil for successfully using a geographical metaphor. The Services Sphere's inherent structure, which results from many of the services being associated with physical entities, makes geographical metaphors a natural fit. That is, if the metaphor mimics the real geography of the objects. The use of such a metaphor can increase user understanding of the Services Sphere to a greater extent possible than for Information Space. Although it should of course be said that such a metaphor is not mandatory, it is merely an option that in certain situations[262] can enhance user understanding.

The relation between physical and virtual geographies is to a large extent dependent upon the service involved. Some services have a direct association with a particular physical object, while others have a link that is purely semantic. An example from the study of the former is the digital camera service as provided by a digital camera. An example of the latter is the text editing service, which is not provided by a physical object, but has a semantic relation with an office. Sometimes the association between a geographic location and a service is a user preference, for example where people put clocks in their houses. The division between those types of association is not so much a strict dichotomy as a sliding scale dependent upon both the service and situation involved.

Geographical metaphors are more promising than no metaphors in the context of navigating services, but only if the relation between the physical and virtual geography is clearly understood by the user.

### 9.1.3. Hypothesis 3: Metadata requirements for services

The third hypothesis is valid within the scope of this research; the metadata requirements for services are different than those for documents when implementing different navigational interfaces. The more complex nature of services means that

---

[262] Situations where the collections of services contain enough ties to physical reality.

often more metadata is required, and it in turn can both differ between users, and take many forms. The tables below give a detailed overview of the metadata requirements that were identified during the research for this thesis. It only concerns metadata that is needed to implement the different navigational interfaces; different uses of services might have different metadata requirements[263]. Most of the requirements can be categorised using the hierarchy of navigation[264], but a limited number of variables within each layer of the hierarchy of navigation has been subject of study[265], which means that the metadata requirements are limited by the same scope. Whether different variables introduce different metadata requirements can be the topic of more extensive studies.

At the bottom level of the hierarchy of navigation are the requirements for organising services or documents. Services and documents can be organised in a variety of ways and each way requires different metadata. See the table below for the requirements identified by this research at the organisation level. The requirements in the table are for a time-based and a location-based organisation, but other ways of organisation will result in different metadata requirements.

| Organisation | Metadata documents | Metadata services |
|---|---|---|
| **Time** | - Last time viewed<br>- Last time modified | - Last time used<br>- Last time changed preferences |
| **Location** | - Semantic connection with a location | - Location of the device providing the service[266]<br>OR<br>- Semantic connection with a location |

The next level up in the hierarchy of navigation is the metaphor level. Each metaphor has its metadata requirements from the documents and services. Since services, when compared to documents, afford different metaphors the metadata requirements differ correspondingly. For the geographical metaphor, which was subject of this research, the differences in metadata requirements are:

263 An example could be metadata for software agents to autonomously work with the service.

264 See section 3.3.

265 For instance, only two types of organisation have been the topic of research.

266 If the service has a clear connection with a device, e.g. is provided by it.

| Metaphor | Metadata documents | Metadata services |
|---|---|---|
| **Geography** | Geographical location | Geographical location |

The interface mode level lies on top in the hierarchy of navigation. The metadata required from services and documents changes with the different interface modes:

| Mode | Metadata documents | Metadata services |
|---|---|---|
| **Text** | | |
| **2D** | | - Service specific 2D icon |
| **3D** | - Coordinates where to position 2D icon | - Service specific 3D icon<br>- Coordinates where to position 3D icon |
| **Speech** | - Sound icons of all metadata | - Sound icons of all metadata |
| **All modes** | - Name<br>- Document type<br>- Access rights<br>- Owner | - Name<br>- Description<br>- Access rights<br>- Security information<br>- Service provider<br>- User preferences<br>- System requirements |

Besides the requirements further issues regarding metadata were identified during the research. When developing services or a service browser the problems arising from how to associate metadata with a service, and problems whether to change it automatically or by hand, will have to be tackled. Depending upon the service involved the association between the two can be a job either for the service itself, another service, or for the service browser. The task is likely to be made even more difficult because while standards are available for associating metadata with documents on the Web, such standards are lacking for services.

Another issue that came up during the research is that metadata can be user specific, where the user doesn't have to be human. Different types of users need different types of metadata. For instance a human might require a well-written description of the service where a software agent might require that information to be in binary format.

A question that came up during research is how, and indeed to what extent, metadata can be changed once it has been associated with a service. While standards for associating and updating metadata for documents are devised by projects such as the Semantic Web project, the same does not apply to services, which makes it a potential direction for future research.

### 9.1.4. The prototype

The Services Platform, the browsers, and the services were not only vehicles for answering the research question by implementing the scenarios and doing user tests, the process of developing each of them was an important element of the exploratory side of this thesis. As was stated at the beginning of this thesis the goal was not only to attempt to test the validity of the hypotheses, but also to undertake a technical exploration that would result in a prototype. The fruits of this exploration have been documented in the relevant chapters (6 and 7) of the thesis. The end result is the Services Platform, with multiple interfaces[267] (browsers), and various functional services all of which can be used for further studies[268].

The Services Platform is the combination of technologies needed to give users access to distributed services. It underlies the navigation layer in the services architecture (see Figure 6-14). The Services Platform enables the delivery and organisation of distributed services that have multiple interfaces.

The browsers are implementations of the navigation layer, which in turn is built on top of the Services Platform (see Figure 6-14). Where the Services Platform is meant to bring users and services together, the browser is meant to navigate the collection of services (the Services Sphere), similar to how a Web browser can be used to navigate the Web (an Information Space). Together with the Services Platform the browser covers all layers in the hierarchy of navigation. Multiple user interfaces were developed for the browser; a text, 2D, 3D, and speech interface.

The services that were developed are compatible with the Services Platform and can be navigated using the service browser. Many of the services are fully functional and can be used for further studies or outside the research realm.

From a user-perspective the Service Platform and its associated browsers offer the unique functionality of navigating distributed services in a user-friendly manner without having to install or update any additional software. New services on the network will automatically be available and the right interface for the client device will automatically be picked out. Service providers can update their services without

[267] A text-based interface, two 2D interfaces, one 3D interface, and a speech interface.

[268] All of which can be found on the accompanying CD-ROM.

requiring the service consumers to install an update. The new version will automatically be used. In short; the burden of maintaining computer systems is taken off the shoulders of users. Altogether it's a step in the direction of the mantra: any service, any device, anywhere, in a user-friendly way and without system administration.

## 9.2. Future work

The initial exploratory nature of this thesis coupled with its broad scope meant that many issues were identified that can serve as a topic for future research. This made it harder to maintain a focus upon a small subset of the questions that arose. Attempting to answer these questions frequently resulted in raising further questions, which was not as problematic as it may seem since one goal of the work done here was to clarify the issues. In the paragraphs below, I address some of the most promising directions and provide concrete research questions.

What is most needed at this point is research that takes the initial exploration of this research further by conducting quantitative studies. It would not only involve large user studies, but also larger test environments that would approach a more realistic situation. The result would answer the question whether the indications as identified here were correct. But besides quantifying the indications found here, other opportunities for future research arose.

This thesis indicated that becoming confused and lost is likely to be a problem when navigating the Services Sphere, just as it is while navigating Information Space. One way of diminishing this might be found in Lynch's[269] guidelines for constructing easily navigable cities. Lynch's guidelines were applied extensively, with varying success[270], in constructing Information Spaces in the hope of solving the problem of users becoming lost. A fruitful research approach would be to try to apply Lynch's findings to navigating the Services Sphere.

*Question 1: Do Lynch's guidelines translate to navigating the Services Sphere, and can they help solve some of the problems of users becoming lost?*

A Services Sphere as defined for the purposes of this thesis consists purely of services. While this was expedient for answering the research questions, it is not a very realistic approach, as the environment through which users navigate will also include documents. Follow-up research could therefore usefully study an environment combining both Information Spaces and the Services Sphere. It could well be that the

---

[269] See (Lynch, 1960) as well as section 3.1 for the discussion.

[270] See (Dieberger, 1994) as well as section 3.1 for the discussion.

answers to the questions stated in this research are different for a hybrid of the Services Sphere and Information Space.

*Question 2: Do the conclusions from this study still hold for an environment consisting of both documents and services?*

During the development of the prototype browsers it surfaced that navigating between services could be done either by using links *inside* other services, or by using a browser that provided the user with an overview of the services available. The literature review showed that a similar division exists with navigating Information Spaces; research in Information Visualization is mainly concerned with visualizing large quantities of information so they can be navigated more easily, while hypermedia studies are mainly concerned with navigating between documents by going from the content of one document to the other. The research done here could only allude to the possibilities of both approaches for navigating services; it neither could nor did go into detail as to which strategy would ensure better usability. It did hint that it might depend upon the situation, the user, and the service.

*Question 3: Do users prefer to navigate between services by using browser that gives an overview of services, or by using links from within each service?*

The link between services and physical objects offers opportunities to navigate services *in a virtual environment,* which was the approach adopted here. But in theory it also means that services can be navigated by navigating the real physical world. Navigating Information Space in this way has already been topic of research[271], but such research has not as yet been conducted into navigating the Services Sphere. This is the point at which the topic of this research clearly intersects with the area of ubiquitous/pervasive computing, and follow-up research could constructively extend the results of this research further into that area.

*Question 4: Can physical reality act as an effective interface to virtual services?*

The prototype offered users the option to start with a service and find the content for the service to act upon, or *vice versa*. The user study could not identify a significant general preference; a more thorough research, preferably with larger user studies, is needed for that.

---

[271] Coincidently while writing this paragraph Nielsen's latest newsletter on the exact same topic arrived (Nielsen, 2002), showing that it is still a relevant topic.

*Question 5: Do users prefer to start with a service and find the content to act upon, or vice versa?*

One of the results of the thesis is the conclusion that services with an actual geographical location provide good opportunities for using geographical metaphors. The prototype and the test assumed that most of the services had such geography, but did not experiment with a situation in which such services were proportionately less. This conclusion raised the question how big the share of services with an actual geographical location should be before a geographical metaphor becomes useful.

*Question 6: How big should the share of services with an actual geographical location be before a geographical metaphor becomes useful?*

Users turned out to have the same expectations from navigating the Services Sphere, as they have from navigating the Web, which led to some problems. Research into the question of how this can be prevented could result in better service browsers.

*Question 7: Which expectations from browsing the Web do users hold when navigating the Services Sphere?*

The study concluded that services have different metadata requirements from documents. The next step would be to development standard formats for those metadata, which are currently non-existent, as the research showed.

*Question 8: What formats are needed for metadata for services?*

This research identified differences in metadata requirements between services and documents when using a time-based or a location-based organisation. Further research should extend the number of ways of organisation and identify the difference in metadata requirements between services and documents for them. The same goes for varying with the metaphor and the mode.

*Question 9: How do the metadata requirements change for yet other ways of organising services, other metaphors, and other modes?*

After identifying the different metadata requirements the question was how to associate the metadata with the service, and how to update it. While standards for associating and updating metadata for documents already exist, this is not the case for services, making it a potentially fruitful direction for future research.

*Question 10: How to associated metadata with a service?*

Assuming that metadata has been associated with a service the problem arises how to update it. As shown in this thesis there can be a large variation in types of metadata for services, and each type of metadata might differ in the need to be kept up to date. The name of the service for example is something that probably won't change very often, but the metadata describing the last time the service was accessed probably will. There has not been any research yet that investigates the possible ways to change metadata associated with services. Although this is largely a technical question, it also brings up interesting questions when looked at from a knowledge technologies point of view. For example the question what metadata *has* to be updated to be useful for semantic navigation.

*Question 11: How, and to what extent, can metadata be changed once it has been associated with a service?*

Yet another opportunity for further research is to put the Services Sphere in a larger perspective than the one used in this thesis. This has been discussed in a preliminary way in (Beute and Larsen, 2002) where ‘audiences’ and ‘institutions’ were added as a further layer at the top of the hierarchy of navigation, and the complete stack was researched as a medium between users and services. Although outside the scope of this research, users and the social constructs in which they live definitely influence the way in which users and services get together. Audiences and institutions are a part of the process of delivering and accessing services, just as they are in the process of communication as studied by mass communication research. Adopting a media-approach would prevent researchers from studying only the technological issues[272] and incorporates media- and other social theories, such an approach might result in a better understanding of the processes involved, and result in a broader theoretical foundation.

*Question 12: How do the social constructs surrounding the process of service provision and consumption influence it?*

Due to the limited scope of this research it was impossible to create large test environments and conduct large user studies. This raises questions about scalability. First of all there is the question whether the results still hold when the amount of services becomes much larger. As said, the collection of services was about thirty, but could in theory be several orders of magnitude bigger. Secondly the question is whether the amount of metadata for services can scale without any problems. Only a limited number of metadata has been topic of this research, although it could have been much more. Thirdly there is the question whether the geographical metaphor scales to much larger virtual environments.

[272] In theory Human-Computer Interaction incorporates these aspects as well, but they are often forgotten or addressed only marginally. They are better addressed by the media-theories.

*Question 13: Do the results of this research scale to environments with larger amounts of services and to services with larger amounts of metadata, and do the indications regarding the geographical metaphor scale to larger virtual environments?*

Besides the above-mentioned issues the prototypes developed for this study (the Services Platform, the browsers, and the services) lend themselves for further experimentation as well.

- **More services.**
  Creating a larger Services Sphere will be necessary for qualitative studies. Although the number of services used with the user-tests was quite large it was nowhere near the number of documents users navigate when, for instance, browsing the Web. Follow-up research could investigate whether the conclusions from this thesis still hold.
- **Dynamic Services Sphere.**
  Only a few services appeared and disappeared during the user-tests (most notably the digital camera that was connected to the network). It is possible with the Service Platform that a large part of the services is (dis) appearing dynamically. It is even likely that there will be real world situations where many services are constantly (dis) appearing, for example in a network where many devices (offering services) are connected to and removed from the network again. This raises the question how this dynamic property of the Services Sphere affects the navigational task.
- **Multiple interfaces per service.**
  Providing services with multiple interfaces means that the same services can be used from a wider range of devices, which affects the navigational task. A limited number of such services has been developed for this study, but experiments with a Services Sphere consisting of more of those services might yield new insights.
- **Test browsers on different devices.**
  The user-tests were done using a personal computer, but the browsers and services are not limited to such a platform. Running the browser and services on different platforms to see whether that affects the navigational task can be an interesting approach for future research.

The answers to the research questions and the Service Platform developed in combination with the service browsers and services, all lend themselves to further study and experimentation. Whether such follow-up research would consist of a larger user study to confirm or deny results presented here, experimentation with more navigational interfaces, making comparisons with other technologies, is irrelevant.

The work done for this thesis is available[273] and can be used as a basis for further effort.

## 9.3. Concluding remarks

Before finishing this thesis I feel obliged to put the research it presents into perspective. The results can be used as guidelines for developing better ways of navigating services but it remains to be seen if the industry will 'pick up on it' and use it. This is a problem often faced by researchers. While it is not specific to the software industry, it is very obvious there. Good research takes time, and one of the defining characteristics of the software industry is a lack of time. Technology changes so quickly that there is rarely any time for reflection or thorough problem research. The industry constantly seeks *immediate* solutions, making that radical changes of practice are almost never an option[274] and most developments and solutions are at best incremental.

A case in point is SMS messaging. Its initial purpose was to notify GSM mobile phone users they had a voice mail message. But soon customers started to use it for person-to-person messaging. The telephone manufacturers did not anticipate this use, (although they certainly welcomed it,) and had to modify their telephones to enhance text entering and legibility. This process was incremental; manufacturers added text-completion capabilities, and increased the screen size somewhat. Apparently there was a large demand for instant messaging, but rather than conducting research to see what would be a good way to simultaneously boost usability and meet demand, the industry chose for quick revenue by extending existing technologies. This was, and remains, an understandable decision from a business point of view. Unfortunately the result was that users had to use a sub-optimal product. In an online discussion Nielsen[275] explained what it was that made SMS a success by comparing it to WAP:

> *SMS is more successful than WAP because it has higher utility even though it has almost as low usability. I usually promote usability, but truth be told, a better model is to analyse the usefulness of a service which is a combination of two parameters:*

---

[273] See the CD-ROM that accompanies this thesis.

[274] In 'The Innovators Dilemma' Clayton M. Christensen (Christensen, 1997) gives some compelling examples of radically new technologies that *did* break with the status quo *and* became a success nonetheless. Proving that it can be done.

[275] See his site: www.useit.com.

> ***Utility****: what does the service do; how closely does it match users' needs.*
> ***Usability****: how easy, efficient, and pleasant is it to use the service*
>
> *Both are necessary: If a system has no utility, it doesn't matter how easy it is to learn or how efficient it is to use. But even the highest utility is for naught if users cannot figure out how to use it*[276]*.*

If the demand for instant messaging using a portable device had been discovered by research, and the telephone companies had had enough time, they could have conducted a usability study that would have revealed the problems associated with sending SMS messages using a mobile telephone. Had that been the case they could have designed a device that had as much *usability* as its undoubted *utility*. Such a situation would have been better for users. Almost no usability studies were conducted and users ended up with sub-optimal product. Thus rather than leading the way research followed it.

The same could well apply the field of which this research is part. It provides some answers and insights regarding service navigation, but many of these require a break with existing practices. At the same time there are already technologies available on the market that try to solve the same problems by slightly extending and modifying existing technologies. It is safe to say that there is more involved in technological development than what research may find to be optimal in terms of usability. It is hard to predict what the role of research in that process will be. But research's *raison d'être,* irrespective of whether it is used as a lead, a guide, or not at all, is to facilitate *understanding* of all of the aspects of technological progress.

---

[276] Wireless Application Protocol: A standard for providing mobile information appliances with secure access to text-based Web pages.

# 10. References

Abowd, Gregory D., Brumitt, Barry, and Shafer, Steven, 2001, Ubicomp 2001: ubiquitous computinginternational conference, Atlanta, Georgia, USA, September 30-October 2, 2001 : proceedings: Berlin etc., Springer.

American Heritage, 1996, American Heritage: Dictionary of the English language: Houghton Mifflin Company.

Ark, Wendy, 1997, Representation matters: the effect of 3D objects and a spatial metaphor in a graphical user interface: Yorktown Heights, N.Y., IBM Research Division.

Ark, Wendy, Dryer, D. C., Selker, Ted, and Zhai, Shumin, 1997, Representation Matters: The Effect of 3D Objects and a Spatial Metaphor in a Graphical User Interface: Proc of HCI'98, 209-219.

Ark, Wendy, 1998, Landmarks to aid navigation in a graphical user interface: Yorktown Heights, N.Y., IBM Research Division.

Arnold, K., Gosling, J., and Holmes, D., 2000, The java programming language: Addison-Wesley.

Arnold, Ken, O'Sullivan, Brian, Waldo, Jim, Wollrath, Ann, and Scheifler, Robert W., 1999, The Jini Specification.

Arthur, Lowell Jay, 1991, Rapid evolutionary development :requirements, prototyping & software creation: New York, Wiley.

Azuma, R. T., 1995, A survey of augmented reality: Presence, 355-385.

Bannon, Liam and Schmidt, Kjeld, 1989, CSCW :four characters in search of a context: Aarhus, Denmark, Aarhus University, Computer Science Dept..

Barreau, D. K. and Nardi, B., 1995, Finding and reminding: File organization from the desktop: ACM SIGCHI Bulletin, 27, 39-43.

Basili, Victor R., 1992, Software modeling and measurement :the goal/question/metric paradigm: College Park, Md., University of Maryland.

Benner, K., Feather, M., Johnson, W., and Zorman, L., 1993, Utilizing Scenarios in the Software Development Process: Information System Development Process, 117-134.

Benyon, D. Navigating information space. 1995. 1st ERCIM Workshop "Towards User Interfaces for All: Current Efforts and Future Trends"30-31 October 1995, ICS-FORTH, Heraklion, Crete, Greece.
Ref Type: Conference Proceeding

Benyon, D. and Höök, K., 1997, Navigation in information space: supporting the individual: Human-Computer Interaction.

Benyon, D. R., 1998, Cognitive ergonomics as navigation in information space: Ergonomics, 41, 153-156.

Berners-Lee, T., Hendler, J., and Lassila, O., 1 A.D., The Semantic Web: Scientific American.

Berners-Lee, T. Information Management: A Proposal. 1989. CERN.
Ref Type: Report

Berners-Lee, T., 1997, World-wide computer: Communications of the ACM, 57-58.

Berners-Lee, Tim and Fischetti, Mark, 1999, Weaving the Webthe original design and ultimate destiny of the Worl Wide Web: New York, N.Y., HarperSanFrancisco.

Beute, B. The third dimension - The logical next step in the evolution of Interfaces? 1999. Rijksuniversiteit Groningen.
Ref Type: Thesis/Dissertation

Beute, B. and Larsen, J. E. Service-oriented media. 2002.
Ref Type: Unpublished Work

Boardman, R. Activity-centric social navigation. 2000. Conference Companion of OZCHI, Sydney.
Ref Type: Conference Proceeding

Brain, M. How ASPs work. http://biz.howstuffworks.com/asp.htm . 2002.
Ref Type: Electronic Citation

Budde, R. and Bacon, Philip, 1992, Prototyping :an approach to evolutionary system development: Berlin, Springer-Verlag.

Burkey, C. Environmental interfaces: HomeLab. 2000. CHI2000.
Ref Type: Conference Proceeding

Bush, V., 1945, As we may think: The Atlantic Monthly, 202-208.

Bylund, M. and Waern, J., 1998, Service contracts: coordination of user-adaption in open service architectures: Personal Technologies, 2.

Bylund, M. and Waern, J., 2001, Personal service environments - Openness and user control in user-service interaction: SICS Technical report, T2001.

Bylund, M., 2001, sView - Architecture overview and system description: SICS Technical report, T2001.

Card, Stuart K., Mackinlay, Jock D., and Shneiderman, Ben, 1999, Readings in information visualization: using vision to think: San Francisco, Calif, Morgan Kaufmann Publishers.

Carroll, J. M., 1995, Scenario-Based Design: Envisioning Work and Technology in System Development: New York, John Wiley & Sons.

Carroll, John M. and Rosson, Mary Beth, 1991, Getting around the task-artifact cycle :how to make claims and design by scenario: Yorktown Heights, N.Y., IBM T.J. Watson Research Center.

Carroll, John M., 2000, Making use :scenario-based design of human-computer interactions: Cambridge, Mass., MIT Press.

Carroll, John M., 2002, Human-computer interaction in the new millenium: New York, New York, ACM Press.

Carzaniga, A., Picco, G. C., and Vigna, G. Designing Distributed Applications with Mobile Code Paradigms. 1997. Proceedings of the 19th International Conference on Software Engineering.
Ref Type: Conference Proceeding

Caswell, D. and Debaty, P. Creating web representations for places. http://www.cooltown.hp.com/dev/wpapers/placeman/placesHUC2000.pdf . 2002.
Ref Type: Electronic Citation

Chalmers, M., 1999, Informatics, Architecture and Language, *in* Social Navigation in Information Space (Munro, A., Hook, K., and Benyon, D., eds.): Springer.

Chen, C., 2000, Individual Differences in a Spatial-Semantic Virtual Environment: Journal of the American Society of Information Science, **51**, 529-542.

Chen, H., Chakraborty, C., Xu, L., oshi, A., and Finin, T. Service Discovery in the Future Electronic Market. 2001. Proc. Workshop on Knowledge Based

Electronic Markets.
Ref Type: Conference Proceeding

Chin, J. P., Diehl, V. A., and Norman, K. L. Development of a Tool Measuring User Satisfaction of the Human-Computer Interface. 1988. SigChi'88.
Ref Type: Conference Proceeding

Christensen, Clayton M., 1997, The innovator's dilemma :when new technologies cause great firms to fail: Boston, Mass., Harvard Business School Press.

Claessens, J., Preneel, B., and Vandewalle, J., 2002, A Tangled World Wide Web of Security Issues: First Monday, 7.

Coad, Peter and Yourdon, Edward, 1990, Object-oriented analysis: Englewood Cliffs, N.J, Yourdon Press.

Cole, T. W., 2002, Creating a Framework of Guidance for Building Good Digital Collections: First Monday, 7.

Columbia University, 1999, The Columbia Encyclopedia: Columbia University Press.

Computer Desktop Encyclopedia, 1998, Computer Desktop Encyclopedia: The Computer Language Company Inc..

Conklin, J., 1987, Hypertext: an introduction and survey: IEEE Computer, 20.

Cook, T., 1994, Do you know where your data are?: Technology Review.

Cooper, Alan, 1995, About face :the essentials of user interface design: Foster City, Calif., IDG Books Worldwide.

Curl cooperation. Consuming web services with the surge environment. 2002. Curl cooperation.
Ref Type: Report

CyberAtlas. Wireless Devices Continue to Proliferate. http://cyberatlas.internet.com/markets/wireless/print/0,,10094_950001,00.html . 2002. INT Media Group.
Ref Type: Electronic Citation

Darken, Rudolf P. Wayfinding in large-scale virtual worlds. 1996. The George Washington University.
Ref Type: Thesis/Dissertation

Deutsch, P. The eight fallacies of distributed computing. Larsen, J. E. 2001.
Ref Type: Internet Communication

Dey, A., Abowd, G. D., and Wood, A. CyberDesk: a framework for providing self-integrating context-aware services. 1998. Proceedings of IUI '98.
Ref Type: Conference Proceeding

Dieberger, A. Navigation in Textual Virtual Environments using a City Metaphor. 1994. Vienna University of Technology, Faculty of Technology and Sciences.
Ref Type: Thesis/Dissertation

Dieberger, A. Providing spatial navigation for the World Wide Web. 1995. International Conference on Spatial Information Theory, COSIT-95 Semmering (Austria).
Ref Type: Conference Proceeding

Dillon, Ronna F. and Schmeck, Ronald R., 1983, Individual differences in cognition: New York, Academic Press.

Dix, A. The myth of the infinite fast machine. 215-228. 1987. Cambridge University Press. Proceedings of HCI'87. D.Diaper & R.Winder.
Ref Type: Conference Proceeding

Dix, A. Design of user interfaces for the web. 1999. Springer. Proceedings of User Interfaces to Data Intensive Systems.
Ref Type: Conference Proceeding

Dix, A., Ramduny, D., Rodden, T., and Davies, N., 2000, Places to stay on the move - software architectures for mobile user interfaces: Personal Technologies, 4.

Dodani, M. H., Hughes, C. E, and Moshell, J. M., 2002, Separation of powers: Object-oriented user interfaces promise evolutionary -rather than revolutionary-upgrades: Byte, 14, 255-262.

Dourish, P. and Chalmers, M. Running out of space: models of information navigation. HCI'94. 1994.
Ref Type: Conference Proceeding

Dourish, P., Bentley, R., Jones, R., and MacLean, A. Getting some perspective: using process descriptions to index document history. 1999. Proceedings of GROUP '99.
Ref Type: Conference Proceeding

Dourish, P., Edwards, W. K., LaMarca, A., and Salisbury, M., 1999, Presto: an experimental architecture for fluid interactive document spaces: ACM Transactions on Computer-Human Interaction, **6**.

Downs, Roger M. and Stea, David, 1973, Image and environment;cognitive mapping and spatial behavior: Chicago, Aldine Pub. Co..

Eckel, B., 2000, Thinking in java.

Eisner Gillett, S., Lehr, W. H., Wroclawski, J. T., and Clark, D. D. A Taxonomy of Internet Appliances. 2000. Telecommunications Policy Research Conference.
Ref Type: Conference Proceeding

Erskine, L. E., Carter-Tod, D. R. N., and Burton, J. K., 1997, Dialogical techniques for the design of web sites: International Journal of Human-Computer Studies, 47.

Fertig, S., Freeman, E., and Gelernter, D., 1996, "Finding and reminding" reconsidered: ACM SIGCHI Bulletin, 28, 66-69.

Forsberg, M., Hook, K., and Svenson, M. Design Principles for Social Navigation Tools. Proceedings of the Workshop on Personal and Social Navigation in Information Space. 1998.
Ref Type: Conference Proceeding

Freeman, E. and Fertig, S., 1995, Lifestreams: Organizing your electronic life: AAAI Fall Symposium: AI Applications in Knowledge Navigation and Retrieval.

Freeman, Eric Thomas, The Lifestreams Software Architecture.

Fries, K. Experiences with Microsoft Bob. Wexelblat, A. and Maulsby, D. 1995. Workshop on Lifelike Computer Characters.
Ref Type: Report

Gamma, Erich, 1995, Design patterns :elements of reusable object-oriented software: Reading, Mass., Addison-Wesley.

Gelernter, D., Fertig, S., and Freeman, E. Lifestreams: An Alternative to the Desktop Metaphor. Proc. CHI'96. 1996.
Ref Type: Conference Proceeding

Gelernter, David Hillel, 1991, Mirror worlds, or, The day software puts the universe in a shoebox: how it will happen and what it will mean: New York, Oxford University Press.

Gelernter, David Hillel, 1998, Machine beauty: elegance and the heart of technology: New York, Basic Books.

Gelernter, David Hillel, 1998, Machine beauty :elegance and the heart of technology: New York, Basic Books.

Gibbs, W. W., 2002, Autonomic Computing: Scientific American.

Gifford, D. K., Jouvelot, P., and O'Tool, J. W., 1999, Semantic file systems: 13th ACM Symposium on Operating Systems Principles.

Gilder, J., 2000, TELECOSM: How Infinite Bandwidth will Revolutionize Our World.

Gong, L. Java 2 Platform Security Architecture. 1997.
Ref Type: Report

Gorny, P. and Tauber, Michael J., 1990, Visualization in human-computer interaction :selected contributions: Berlin, Springer-Verlag.

Gödel, Kurt, 1962, On formally undecidable propositions of Principia mathematica and related systems: Edinburgh and London, Oliver and Boyd.

Grønbæk, K. and Trigg, R. H. Toward a Dexter-based Model for Open Hypermedia: Unifying Embedded References and Link Objects. 149-160. 1996. ACM Conference on Hypertext.
Ref Type: Conference Proceeding

Haartsen, J., Naghshineh, M., Inouye, J., Joeressen, O., and Allen, W., 1998, Bluetooth: Vision, Goals, and Architecture: Mobile Computing and Communications Review.

Halasz, F. G. and Schwartz, M., 1990, The Dexter hypertext reference model: Communications of the ACM.

Halls, David A., 1997, Applying mobile code to distributed systems: Cambridge Cambridgeshire, University of Cambridge, Computer Laboratory.

Hamfors, O. Service designer: Lets the user create her own user-interface to web services. 2001.
Ref Type: Thesis/Dissertation

Hamit, F., 1993, Virtual reality and the exploration of cyberspace: Sams Publishing.

Henderson, D. A. and Card, S. K., 1986, Rooms: The use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface: ACM Transactions on Graphics, 5, 211-243.

Henshall, John and Shaw, Sandy, 1988, OSI explained :end-to-end computer communication standards: Chichester, West Sussex, England, E. Horwood.

Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, and Verplank, 2002, Curricula for Human-Computer Interaction: ACM SIGCHI.

Hodges, T. and Katz, R. Enabling 'smart spaces': Entity description and user interface generation for a heterogeneous component-based distributed system. 1998. DARPA/NIST. Smart Spaces Workshop.
Ref Type: Conference Proceeding

Hofstadter, Douglas R., 1979, Gödel, Escher, Bach
an eternal golden braid: New York, Basic Books.

Horton, William K., 1994, Designing and writing online documentation :hypermedia for self-supporting products: New York, Wiley.

Hu, P. J., Ma, P., and Chau, P., 1999, Evaluation of user interface designs for information retrieval systems: A computer-based experiment: Decission Support Systems, 27, 125-143.

Husemann, D., 2001, Pervasive computing: Hogwarts, StarTrek, reality and back: Computer Networks.

IBM. Web Services Browser. 2002. IBM.
Ref Type: Computer Program

IBM Web Services Architecture Team. Web Services architecture overview. 2000.
Ref Type: Report

International Data Corp. ASPs Look to Rebound from Slump. http://cyberatlas.Internet.com/big_picture/applications/article/0,,1301_1001741,00.html . 2002.
Ref Type: Electronic Citation

Ishii, H. and Ullmer, B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. 1997. MIT. Proceedings of CHI '97.
Ref Type: Conference Proceeding

James, William, 1890, The principles of psychology: New York, Henry Holt & company.

Johnson, Steven, 1997, Interface Culture: How new technology transforms the way we create and communicate: Harper Collins.

Jose, R. and Davies, N. Scalable and Flexible Location-based Services for Ubiquitous Information Access. 1999. Proceedings of the First International Symposium on Handheld and Ubiquitous Computing.
Ref Type: Conference Proceeding

Jul, S. and Furnas, W. Navigation in Electronic Worlds. CHI'97. 1997.
Ref Type: Conference Proceeding

Kandogan, E. and Schneiderman, B. Elastic Windows: Evaluation of Multi-Window Operations. Proceedings of Human Factors in Computing Systems (CHI 97). 1997. Proceedings of Human Factors in Computing Systems (CHI 97).
Ref Type: Conference Proceeding

Kazman, Rick, 1995, Scenario-based analysis of software architecture: Waterloo, Ont., Canada, University of Waterloo, Computer Science Dept..

Kindberg, T., Barton, J., and Jeff Morgan, Gene Becker Debbie Caswell Philippe Debaty Gita Gopal Marcos Frid Venky Krishnan Howard Morris John Schettino Bill Serra Mirjana Spasojevi. People, Places, Things: Web Presence for the Real World. 2000. WMCSA 2000.
Ref Type: Conference Proceeding

Kirsch, J. Knowledge Technologies Workshop. 2002. Luxembourg.
Ref Type: Report

Kucuk, M. E., Olgun, B., and Sever, H., 2000, Application of metadata concepts to discovery of Internet resources: Advances in Information Systems.First International Conference, ADVIS 2000, 1909.

Kuhn, W. and Blumenthal, B. Spatialization: Spatial Metaphors for User Interfaces. 1996. Department of Geoinformation, Technical University, Vienna. Geoinfo-Series.
Ref Type: Report

Lakoff, George and Johnson, Mark, 1980, Metaphors we live by: Chicago, University of Chicago Press.

Larsen, J. E. and Beute, B. Towards Services Platforms. 2001.
Ref Type: Unpublished Work

Laubacher, R. J. and Malone, T. D., 1997, Two Scenarios for 21st Century Organizations: Shifting Networks of Small Firms or All-Encompassing "Virtual Countries"?: MIT, 21st Century Initiative Working Papers.

Laurel, Brenda, 1995, The art of human-computer interface design: Reading, Mass., etc., Addison-Wesley.

Lindenberg, Jasper, Neerincx, Mark A., and Pemberton, Steven, 2001, Support concepts for Web navigation: a cognitive engineering approach: Proceedings of the Tenth International World Wide Web Conference.

Lumbreras, M. F. Hypertext for blind people. Hypertext'93 . 1993.
Ref Type: Conference Proceeding

Lynch, Kevin, 1960, The image of the city: Cambridge Mass., Technology Press.

Malone, Thomas D., 1983, How do people organize their desks? Implications for the design of office information systems: ACM Transactions on Office Systems, 1, 99-112.

Mander, Richard, Salomon, Gitta, and Wong, Yin Yin, 2001, A "Pile" metaphor for supporting casual organization of information: Proceedings CHI '92, 627-634.

Mankoff, J., Somers, J., and Abowd, G. D., 1998, Bringing people and places together with dual augmentation : Paper submitted for review to conference on Cooperative Virtual Environments.

Mattern, F. and Sturm, P. From Distributed Systems to Ubiquitous Computing - The State of the Art, Trends, and Prospects of Future Networked Systems. 2002.
Ref Type: Unpublished Work

Mayhew, Deborah J., 1992, Principles and guidelines in software user interface design: Englewood Cliffs, N.J., Prentice Hall.

Mazuryk, T. and Gervautz, M. Virtual Reality History, Applications, Technology and Future. 1996. Institute of Computer Graphics and Algorithms, Vienna University of Technology. Technical Report TR-186-2-96-06.
Ref Type: Report

McCandless, M. Managing Your Privacy in an Online World. IEEE Intelligent Systems . 1997.
Ref Type: Magazine Article

McIlraith, S., Son, T. C., Zeng, H., and , 2001, Semantic web services: IEEE Intelligent Systems.Special Issue on the Semantic Web, 16, 46-53.

McKnight, Cliff, Dillon, Andrew P., and Richardson, John, 1991, Hypertext in context: Cambridge, Cambridge University Press.

McKnight, Cliff, Dillon, Andrew, and Richardson, John, 1993, Hypertext :a psychological perspective: New York, E. Horwood.

Miller, R. B. Response time in man-computer conversational transactions. 1968. Proceeding AFIPS Fall Joint Computer Conference.
Ref Type: Conference Proceeding

Mohnkern, Kenneth E. and Carnegie Mellon University.School of Design., 1997, Affordances, metaphor, and interface design: Pittsburgh, Pa., Carnegie Mellon University.

Munro, Alan J., Höök, Christina, and Benyon, David, 1999, Social Navigation of Information Space: London, Springer.

Myers, B., Hollan, J., and Cruz, I., 1996, Strategic directions in Human-Computer Interaction: ACM Computing Surveys, 28.

Nardi, B. A. and Barreau, D., 1997, "Finding and Reminding" Revisited: Appropriate Metaphors for File Organization at the Desktop: ACM SIGCHI Bulletin, 29, 76-78.

Neerincx, Mark A. and Pemberton, Steven. Support concepts for web navigation: a cognitive engineering approach. 2002. ACM. WWW10.
Ref Type: Conference Proceeding

Nelson, T. H., 1967, Getting It Out of Our System, *in* Information Retrieval: A Critical Review (Schechter, G., ed.).

Neumueller, M. Hypertext Semiotics in the Commercialized Internet. 2001.
Ref Type: Thesis/Dissertation

Nielsen, J. Finding usability problems through heuristic evaluation. Human Factors in Computing Systems . 1992.
Ref Type: Conference Proceeding

Nielsen, J., 1992, Evaluating the thinking aloud technique for use by computer scientists, *in* Advances in Human-Computer Interaction (Hartson, H. R. and Hix, D., eds.): Norwood, Ablex.

Nielsen, J., 1994, Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier, *in* Cost-Justifying Usability (Bias, R. G. and Mayhew, D. J., eds.): Academic Press.

Nielsen, J. The Network is the User Experience: Microsoft's *.NET* Announcement. http://www.useit.com/alertbox/20000625.html . 2000.
Ref Type: Electronic Citation

Nielsen, J. WAP backlash. http://www.useit.com/alertbox/20000709.html . 2000.
Ref Type: Electronic Citation

Nielsen, J. Making the Physical Environment Interactive. http://www.useit.com/alertbox/20020805.html . 2002.
Ref Type: Electronic Citation

Nielsen, Jakob, 1991, Hypertext and hypermedia: Boston, Academic Press.

Nielsen, Jakob, 1993, Usability engineering: Boston, Academic Press.

Norman, D., 1998, The invisible computer: Why good products can fail, the personal computer is so complex and information appliances are the solution: Cambridge, MA, MIT Press.

Nylander, S. and Bylund, M. Device independent services. 2002. SICS, Upsala Sweden.
Ref Type: Report

O'Tool, J. W. and Gifford, D. K., 1992, Names should mean what, not where: 5th ACM European Workshop on Distributed Systems.

Odlyzko, A., 1999, The Visible Problems of the Invisible Computer: A Skeptical Look at Information Appliances: First Monday, 4.

Ohta, Yuichi and Tamura, Hideyuki, 1999, Mixed reality :merging real and virtual worlds: Tokyo, Ohmsha.

Oram, Andrew, 2001, Peer-to-peer :harnessing the benefits of a disruptive technology: Beijing, O'Reilly.

Passini, Romedi, 1984, Wayfinding in architecture: New York, Van Nostrand Reinhold.

Penrose, Roger, 1989, The emperor's new mind
concerning computers, minds, and the laws of physics: New York, Oxford University Press.

Peponis, J., Zimring, C., and Choi, Y. K., 1990, Finding the Building in Wayfinding: Environment and Behavior, 22, 555-590.

Pinker, Steven, 1997, How the mind works: New York, Norton.

Plaisant, C. and Schneiderman, B. Organization overviews and role management: Inspiration for future desktop environments. IEEE Proc. 4th Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises. 1995.
Ref Type: Conference Proceeding

Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., and Schneiderman, B., 1998, LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records: 1998 American Medical Informatic Association Annual Fall Symposium (Orlando, Nov.9-11, 1998), 76-80.

Postel, J. Telnet protocol. 1972.
Ref Type: Report

Pradhan, S. Semantic location. http://www.cooltown.hp.com/dev/wpapers/semantic/semantic.asp . 2002.
Ref Type: Electronic Citation

Preece, Jenny, 1994, Human-computer interaction: Wokingham, England, Addison-Wesley Pub. Co..

Preece, Jenny, Rogers, Y., and Sharp, H., 2002, Interaction design: beyond human-computer interaction: Wiley & Sons.

Ramasami, V. C. Security, Authentication And Access Control For Mobile Communications. 2002.
Ref Type: Report

Ramsay, M. and Nielsen, J. WAP usability report. 2000.
Ref Type: Report

Redström, J., Holmquist, L. E., Dahlberg, P., and Ljungstrand, P. Ad Hoc Information Spaces. 1999.
Ref Type: Unpublished Work

Reeves, Byron and Nass, Clifford Ivar, 1996, The media equation :how people treat computers, television, and new media like real people and places: Stanford, Calif., Center for the Study of Language and Information.

Rekimoto, J., 1999, Time-Machine Computing: A Time-Centric Approach for the Information Environment: {ACM} Symposium on User Interface Software and Technology, 45-54.

Rischpater, Ray, 2001, Internet appliances: New York, Wiley.

Robertson, G. and Roseman, M. Using a room metaphor to ease transitions in groupware. 1998. Research report 98/611/02, Department of Computer Science, University of Calgary.
Ref Type: Report

Robertson, G., Dantzich, M. van, Robbins, D., Czerwinski, M., Hinckley, K., Risden, K., Thiel, D., and Gorokhovsky, V. The Task Gallery: a 3D window manager. Proceedings of CHI 2000 . 2000. ACM Press.
Ref Type: Conference Proceeding

Russel, S. J. and Norvig, P., 1994, Artificial Intelligence: A modern approach.

Salton, Gerard and MacGill, Michael J., /19, Introduction to modern information retrieval: New York, N.Y., etc., McGraw-Hill.

Saussure, Ferdinand de, 1959, Course in general linguistics: New York, Philosophical Library.

Schmidt, Kjeld and Bannon, Liam, 1992, Taking CSCW Seriously: Supporting articulation work: Computer Supported Cooperative Work, 1, 7-40.

Schneiderman, B. and Plaisant, C., 1994, The future of graphic user interfaces: personal role managers: People and Computers IX.

Schneiderman, B., 2000, Universal usability: Communications of the ACM, 43, 85-91.

Schomaker, L., Nijtmans, J., and et al. A Taxonomy of Multimodal Interaction in the Human Information Processing System. 1995.
Ref Type: Report

Schwartz, P., 1996, The Art of the Long View : Planning for the Future in an Uncertain World.

Segal, B. A Short History of Internet Protocols at CERN. 1995.
Ref Type: Unpublished Work

Selker, Ted, May, Ian, and Zhai, Shumin, 1997, Representation matters: spatial interface can facilitate target acquisition: Yorktown Heights, N.Y., IBM Research Division.

Shubin, H. and Meehan, M. M. Navigation in web applications. ACM Interactions Magazine 4[6]. 1997.
Ref Type: Magazine Article

Steed, A. A Survey of Virtual Reality Literature. 1993.
Ref Type: Report

Streitz, N. A., Konomi, Shin ichi, Burkhardt, Heinz Jürgen, and Gesellschaft für Mathematik und Datenverarbeitung., 1998, Cooperative buildings :integrating information, organization, and architecture : first international workshop, CoBuild '98, Darmstadt, Germany, February 25-26, 1998 : proceedings: Berlin, Springer.

Sun Microsystems, Inc. Java Speech API Specification. 2001.
Ref Type: Report

Sutcliffe, Alistair, 1995, Human-computer interface design: Basingstoke, Hampshire, Macmillan.

Tanenbaum, Andrew S., 1996, Computer networks: Upper Saddle River, N.J, Prentice Hall PTR.

Tanenbaum, Andrew S. and Steen, Maarten van, 2002, Distributed systems :principles and paradigms: Upper Saddle River, N.J., Prentice Hall.

Theng, Y. L., Thimbleby, H., and Jones, M. 'Lost in hyperspace': Psychological problem or bad design? 1996. APCHI'96, Singapore.
Ref Type: Conference Proceeding

Thimbleby, H., Jones, M., and Theng, Y. L. Is "lost in hyperspace" lost in controversy? 2002.
Ref Type: Unpublished Work

Thorndyke, Perry W., 1980, Spatial cognition and reasoning: Santa Monica, CA, Rand.

Tognazzini, Bruce, 1992, Tog on interface: Reading, Mass., Addison-Wesley.

Tweedie, L. A. Characterizing interactive externalizations. 1997. Proceedings of CHI'97.
Ref Type: Conference Proceeding

Venners, B. ServiceUI draft specification. 2002.
Ref Type: Report

Venners, B. Jini Place API Draft Specification. 2002.
Ref Type: Report

Wack, Pierre, 1985, Scenarios :uncharted waters ahead: Boston, Mass., Harvard Business Review.

Waldo, J., Wyant, G., and Wollrath, A. A note on distributed computing. 1994. Sun Microsystems Laboratories. Sun Microsystems Laboratories Tech Report.
Ref Type: Report

Waldo, J. The end of protocols. 2000.
Ref Type: Internet Communication

Waldo, Jim, 1999, The Jini Architecture for Network-Centric Computing : Communications of the ACM, 42, 76-82.

Waldo, Jim. Network Evolution. http://java.sun.com/features/2001/06/golden.jini.p.html . 2001.
Ref Type: Conference Proceeding

Walker, L. A Visual Rather Than Verbal Future. Washington Post May 9, E01. 2002.
Ref Type: Newspaper

Weiser, M., 1991, The computer for the twenty-first century: Scientific American.

Weiser, M., 1993, Ubiquitous Computing: IEEE Computer.

Wertheim, Margaret, 1999, The pearly gates of cyberspace :a history of space from Dante to the Internet: New York, W.W. Norton.

Wired. Gates Finally Discovers Security. Wired . 2002.
Ref Type: Magazine Article

Wittgenstein, Ludwig, 1953, Philosophical investigations: New York, Macmillan.

Wong, W., Kane, M., and Ricciuti, M. Web services try to rise above high-tech din. 2001. CNET News.com.
Ref Type: Report

Zamperoni, A., Gerritsen, B., and Bril, B. Evolutionary Software Development: An Experience Report on Technical and Strategic. 1995. Technical report Leiden University, department of computer science.
Ref Type: Report

# 11. Appendices

## 11.1. Sample session browser

Here is a sample session of a user using CyberCMD to go to his place with favourite services (HotSpot) and from there to his house and his office. From there he launches some services are requests information about them.

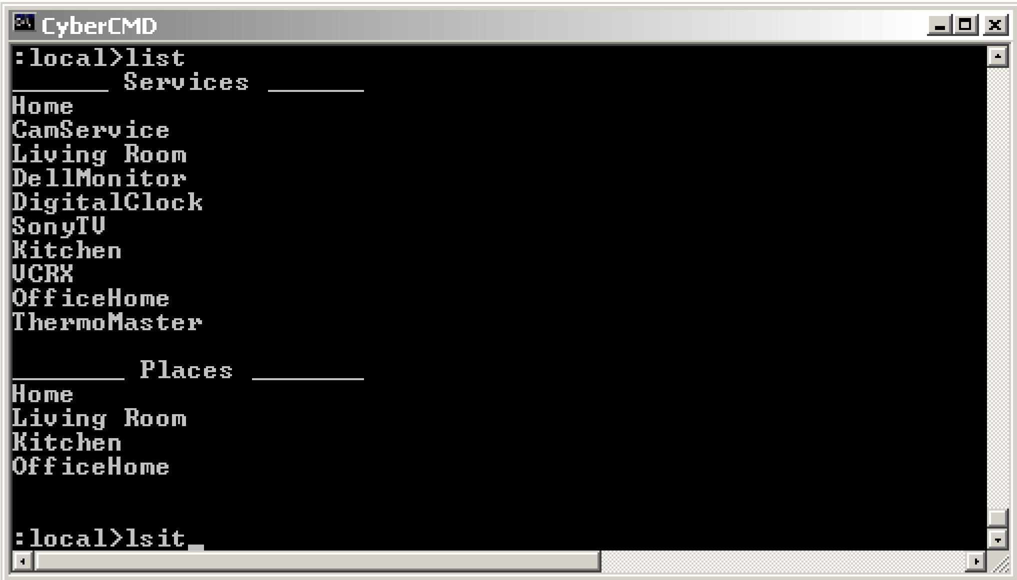



**Figure 11-1 A screenshot of the text-based interface (CyberCMD) of the browser.**

The output to the screen during the session:

```
Searching for local services...please wait 5 seconds...
Found 10 services, type 'list' to see them.
***************************************************
*                    CyberCMD                     *
*          Your Shell for CyberSpace!         v0.2*
***************************************************
* Type 'help' for help                            *
* Type 'go hotspot' to go to your starting place  *
* Type 'commands' for a list of available commands*
* Type 'exit' to quit                             *
***************************************************

:local>list
```

```
______ Services ______
Home
CamService
Living Room
DellMonitor
DigitalClock
SonyTV
Kitchen
VCRX
OfficeHome
ThermoMaster
_______ Places _______
Home
Living Room
Kitchen
OfficeHome

:local>go HotSpot
:local>HotSpot>list
______ Services ______
Home
OfficeCTI
_______ Places _______
Home
OfficeCTI

:local>HotSpot>go Home
:local>HotSpot>Home>list
______ Services ______
Living Room
Kitchen
OfficeHome
ThermoMaster

_______ Places _______
Living Room
Kitchen
OfficeHome

:local>HotSpot>Home>info ThermoMaster
Name: ThermoMaster
Model: 0.10
Version: 0.10
Manufacturer: Jakob Eg Larsen & Berco Beute
Vendor: Center for Tele-Information, DTU
```

```
Serialnr: 1

:local>HotSpot>Home>go OfficeHome
:local>HotSpot>Home>OfficeHome>list
______ Services ______
TextEditor
DellMonitor
DigitalClock

:local>HotSpot>Home>OfficeHome>launch DigitalClock
Retrieving Service...please wait max. 10 seconds...
:local>HotSpot>Home>OfficeHome>go back
:local>HotSpot>Home>go back
:local>HotSpot>list
______ Services ______
Home
OfficeCTI
_______ Places _______
Home
OfficeCTI

:local>HotSpot>go OfficeCTI
:local>HotSpot>OfficeCTI>list
______ Services ______
TextEditor
CamService
DigitalClock
CashConverter

:local>HotSpot>OfficeCTI>launch TextEditor
Retrieving Service...please wait max. 10 seconds...
:local>HotSpot>OfficeCTI>go back
:local>HotSpot>go back
:local>list
______ Services ______
Home
CamService
Living Room
DellMonitor
DigitalClock
SonyTV
Kitchen
VCRX
OfficeHome
ThermoMaster
```

```
_______ Places _______
Home
Living Room
Kitchen
OfficeHome
```

### 11.2. Summarised test results

This section contains the tables with summarised results. A complete listing of all the results can be found in the next section (11.3).

#### 11.2.1. Observations

| General observations | |
|---|---|
| **Observation** | **Count** |
| Users consciously reason where a 'non-physical' service (e.g. a Text Editor Service) could reside, and come up with the right answer (Office). | 6 |
| Users go searching for a clock when asked to check the time, and go to the places where they would normally expect a clock | 6 |
| Users have difficulties understanding the concept of a service's user interface searching for other services to cooperate with (e.g. the interface for the digital camera that can search for displaying services). | 6 |
| The user searches for a central place to control all the lights in the house (which is unavailable) | 3 |
| Users have difficulties understanding they can go both ways when using a service from within another (e.g. when displaying pictures from the digital camera on the TV, the user could start with the camera and find the TV, or *vice versa*). | 2 |
| Confuses reality with the virtual environment they navigate. They try to find a computer in the Services Sphere and then realize they already sit behind one and tries to use that one. | 2 |

| Text-based interface | |
|---|---|
| **Observation** | **Count** |
| The user relies a lot on the help commands in the beginning | 3 |
| The user accidentally tries to use commands from existing command line tools (DOS/UNIX) | 2 |
| After a slow start the user quickly learns to use the interface to navigate around | 6 |

| | |
|---|---|
| The names of the commands are found to be unusual and difficult | 6 |
| Spelling errors are often made and not easily realized | 2 |
| The user is initially confused due to the separation between services and places | 1 |
| The user is confused by the fact that places are listed under services as well | 1 |
| The user would like to automatically see all available places and services listed | 1 |
| The user expects a text-based UI instead of a graphical UI for the services | 1 |
| The user wishes for more metadata about the services to find out what they are before launching them | 1 |
| Users think every service is a place: 'go service' instead of 'go place' | 4 |
| The user tries to launch a place ('launch home') | 1 |
| The user is confused by the long retrieval time, thinks the browser crashed | 1 |
| The users try to go to a place to which there is no link in the place they are in. | 4 |

| **2D interface** | |
|---|---|
| **Observation** | **Count** |
| The user becomes proficient very fast due to familiarity with 2D interfaces | 3 |
| The user makes no more mistakes of trying to go from one place to another while there is no link between them | 3 |
| The user finds the 'history list' confusing | 1 |

| **3D interface** | |
|---|---|
| **Observation** | **Count** |
| The user has to get used to using a keyboard and a mouse for navigating 3D environments | 1 |
| The users find the interface intuitive but too slow. | 2 |
| Users get annoyed by how long it takes to navigate between services | 3 |

| | |
|---|---|
| Users get confused by the abstract space that contains all services that have no ‘physical’ counterpart (MetaVerse) | 3 |
| Users find it irritating that they have to click an object to see if, and if so, what, services it provides. | 2 |
| The user misses a listing of all services in a place | 1 |
| The user wishes for a map of the house to jump directly to a place | 1 |

### 11.2.2. Questionnaires

| **Text-based interface** | |
|---|---|
| **Quality of use** | |
| **Topic** | **Questions** |
| **Satisfaction** | **2,7,12,15,19,21,23,25,28,35,38** |
| No subject found the text-based interface to be *frustrating*, but the results are divided with regards to how much they *enjoyed* using the software. Some enjoyed it, others didn't. None of them found the interface to be attractive though. | |
| **Learning to use the software** | **5,10,26,30,33,36,40** |
| On average none of the subjects indicated to have had many problems in learning the software. | |
| **Feeling in control of the software.** | **6,11,13,17,32** |
| The majority of subjects said to feel in control of the software (17,32), but other questions indicate that such is not always the case (6,11). There is apparently a difference between subjects in how much they feel to be in control. | |
| **Behaviour of the software** | **1,4,9,21,34,37** |
| All subjects thought that the software responded fast enough to their input. The results also show that most users find the behaviour of the software understandable, but some experienced some unexpected behaviour, or behaviour they didn't understand. | |
| **Help** | **3,8,24** |
| The assistance provided by the software was judged to be sufficient by most users. | |

| **General usability** | **14,16,18,20,22,27,29,31,39** |
|---|---|
| All of the subjects found the software to be consistent and it didn't interrupt the way they normally work. They knew how to act upon the information provided and were able to perform the task in a straightforward manner.<br><br>There was some variation with regards to the information was displayed at each stage, only half of the respondents found it to be sufficient. Most did find the organisation of the information on screen logical though.<br><br>There was no unanimous answer to the question whether user needs were taken into account sufficiently. | |
| **Navigation** | |
| **Topic** | **Questions** |
| **Metadata** | **15,16,17** |
| The name a service and the location from where it is accessible are insufficient for all users to judge what a service can do. There was much variation in how easy subjects could find out what a service can do. | |
| **Getting around** | **1,4,6,7,8,9,13,14** |
| On average navigating around was not judged to be very difficult by most subjects, although some had troubles finding out what could be done in one place and getting from one place to another. | |
| **Geographical metaphor** | **10,11,12** |
| Most subjects found the geographical metaphor a useful and intuitive way of organising services. Only one subject was confused by it. | |
| **Becoming lost** | **2,3,5** |
| Becoming lost is a problem for half of the users. | |

| **2D interface** | |
|---|---|
| **Quality of use** | |
| **Topic** | **Questions** |
| **Satisfaction** | **2,6,10,12,16,19,24,31,34** |
| Compared to the other interfaces satisfaction of use was highest for the 2D interface. It was judged to be the most attractive. | |
| **Learning to use the software** | **4,8,22,26,29,36** |
| It was easy for most subjects to learn to perform the tasks using the 2D interface. | |
| **Feeling in control of the software.** | **5,9,14,28** |
| Even though most of the subjects found the 2D interface easiest to use, not all felt in complete control. | |

| | |
|---|---|
| **Behaviour of the software** | **1,21,30,33** |
| The 2D interface behaves as expected, although a few respondents would like it to respond faster. | |
| **Help** | **3,7,20** |
| The help provided seems to be sufficient for most users. | |
| **General usability** | **11,13,15,17,18,23,25,27,32,35** |
| Although most users found that task can be performed in a straightforward manner, and with relative ease, 3 out of 5 respondents didn't feel user needs had been fully taken into account. | |
| **Navigation** | |
| **Topic** | **Questions** |
| **Metadata** | **15,16,17** |
| Users had little problems finding information about a service. | |
| **Getting around** | **1,4,6,7,8,9,13,14** |
| The subjects agreed on the fact that the 2D interface was an easy way to navigate services. | |
| **Geographical metaphor** | **10,11,12** |
| There was no agreement on whether the geographical metaphor was useful with the 2D interface. | |
| **Becoming lost** | **2,3,5** |
| Becoming lost is a problem for one subject. | |

| **3D interface** | |
|---|---|
| **Quality of use** | |
| **Topic** | **Questions** |
| **Satisfaction** | **2,6,10,12,16,20,25,32,35** |
| Satisfaction was lowest with the 3D interface, although most respondents found it to be an attractive interface. | |
| **Learning to use the software** | **4,8,23,27,30,37** |
| Although some users said to have initial problems learning how to navigate, most users thought to have learned to use the software rather quickly. | |
| **Feeling in control of the software.** | **5,9,14,29** |
| Only one subject was sure enough in her use of the software that she never felt she didn't know what to do next. Three subjects didn't feel in command of the software. Four subjects found it hard to make the software do what they wanted it to do. | |

| | |
|---|---|
| **Behaviour of the software** | **1,18,22,31,34** |
| The software is slow and responds slowly according to four respondents. Only one subject found that the software behave in a way that she was expecting. | |
| **Help** | **3,7,21** |
| Help was insufficient for users to get started. | |
| **General usability** | **11,13,15,17,19,24,26,28,33,36** |
| Although the 3D interface was intuitive and easy to learn the general usability was not found to be very good due to the fact that navigating 3D virtual spaces takes a lot more effort than with the other interface modes. | |
| **Navigation** | |
| **Topic** | **Questions** |
| **Metadata** | **13,14,15** |
| For 3 of the subjects it was easy to find out what a service could do. | |
| **Getting around** | **1,4,6,7,8,12** |
| All subjects judged the 3D interface to be inefficient for navigation. It is simple and consistent but costs a lot of effort. | |
| **Geographical metaphor** | **9,10,11** |
| The geographical metaphor was fitting and intuitive according to 3 subjects, but 2 subjects found it to be of limited use. | |
| **Becoming lost** | **2,3,5** |
| Although 4 subjects said they always knew where they were in the system, 2 explicitly stated they became lost at some point. | |

### 11.2.3. Open questions

| |
|---|
| **Do you feel that navigating services is very different from navigating documents (e.g. Web browsing)?**<br>Depends on the interface mode. Similar in 2D mode, but different in 3D mode. In 3D mode navigating documents is more difficult than navigating services.<br><br>**Yes:**<br>Documents are organised by the meaning of their content, and services by their function and the location and situation where they can be used.<br><br>Services are more abstract. It is more difficult to find out what they are for.<br><br>The same naming of places and services can be confusing. It isn't clear what the difference is by just looking at it.<br><br>**No:**<br>It is just like Web browsing.<br><br>Similarities with navigating tree structured documents.<br><br>**If so, what are the most important differences for you?**<br>Documents depend on the context information, services on general knowledge of similar services and concepts of the real world. |
| **Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?**<br>**Text:**<br>Very fast when you know the commands.<br><br>**2D:**<br>Fast to navigate.<br><br>Better overview of available services than with the text-based or 3D interface.<br><br>I'm used to 2D interfaces.<br><br>Easy to learn.<br><br>**3D:**<br>Only to learn a complex service environment. |

**Name the advantage of each types of interface *for navigating services*:**
**Text-based:**
Fast.

Feeling of complete control.

Requires little resources (usable on limited devices).

What you see is what you can use (no ‘useless’ objects like walls, tables etc.)

**2D:**
Speed.

Visually appealing.

Being used to 2D interfaces.

Good overview: many options and services displayed at once.

**3D:**
Fun to use.

Visually appealing.

Good overview of where you are in the system.

Clearly shows relations between services.

Gives good overview of large amounts of services.

Good way to learn a complex service environment for the first time.

Uses the fact that humans are good at remembering the *location* of a service, as well as the route to that location.

**Name the disadvantage of each types of interface *for navigating services*:**
**Text-based:**
Not fun to use.

Difficult to learn.

Typing is laborious.

A name provides too little information of what the service does.

Lack of orientation points. One has to actively wanderer around to see what’s there.

Need for a keyboard, because point-and-click isn’t possible.

Metadata on service is hard to find.

**2D:**
Uses more screen-space then necessary.

| |
|---|
| Lots of mouse-movement needed.<br><br>**3D:**<br>Time it takes to get from one service to another[277].<br><br>Clutters up the interface by showing many objects that provide no service.<br><br>The available services at a place were not 'listed' as with a list of documents in a Web page.<br><br>The 'mental picture' of the place that is navigated is, other than with text-based and 2D interface, pre-constructed and might differ from the users own 'mental picture', which might confuse the user.<br><br>Not suitable for things you *have* to do and where fun is not an option.<br><br>Resource intensive.<br><br>Requires advanced mouse/keyboard skills.<br><br>Expensive and difficult to design. |
| **Do you have suggestions for other types of interfaces that you would like?**<br>Speech-based.<br><br>A combination of text/2D/3D/Speech interfaces. An interface that let's you switch between the modalities; 3D to learn the system, 2D to get things done, and text to get things done fast.<br><br>Augmented reality. |
| **Describe the mental picture of where you were while navigating?**<br>I was picturing a real physical place where the services fitted in.<br><br>With more abstract services I would have to guess more.<br><br>A tree-structure with the text and 2D interface.<br><br>With the 3D interface I was thinking of moving in a picture.<br><br>I had no mental picture with the 3D interface because it was already created for me. |
| **Do you think there are better ways to organize the services? E.g. by time or activity.**<br>No, but it doesn't fit all kinds of services; where would 'joke of the day service' be located?<br><br>**Below the user interface level:**<br>Depends on the service; if it has a direct link with a physical equivalent it is the best way, it is harder for abstract services. |

[277] One subject noted that children with experience in playing computer games might disagree.

| |
|---|
| *Activity*: for abstract services such as banking services.<br><br>*Personal preferences*: enable the user to reorganize the services.<br><br>**At the user interface level:**<br>A tree-based organisation of the services, showing the hierarchy and relation between places at once (on multiple levels). |
| **Is the geographical metaphor more useful with one type of interface compared to the others?**<br>More useful with the text-based and 2D interfaces, since in the 3D browser I do not really find it a metaphor.<br><br>It totally suites the 3D interface, while less the 2D interface and not the command-line interface.<br><br>**Yes:**<br>In the 3D interface I think this is the best way to organize things. It closely matches the knowledge people already have of the world.<br><br>**No:**<br>The difference in functionality is not related to the metaphor. |
| **Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?**<br><br>**Yes:**<br>The 3D interface requires prior knowledge of which services are being offered by which objects.<br><br>3D is the most intuitive, as you immediately see what things are meant to do or be.<br><br>3D gives more visual information and location information. |
| **Can you think of additional ways for services to convey information about what they offer?**<br>Location sets the expectations to go and look for a service and what you think it could do in advance.<br><br>By using icons that are unique for each service. |
| **Comments**<br>The geographical metaphor might be useful, but only up 'till a certain point. At one point the variables that make up the geographical metaphor vary too much to be of any use anymore.<br><br>Let users organise the services themselves, especially useful for the abstract services.<br><br>Make users aware that documents are stored for them somewhere.<br><br>Give users the option to explicitly store their documents where they want to, giving |

| |
|---|
| them the feeling of being in control. This would enable the users to adopt this new way of thinking and working at their own rate.<br>The meaning of abstract services can differ a lot, depending on a person's background, location, culture, experience and so forth.<br>Enable searching for a service in the interface.<br>Provide a text field in which you can enter the first tree characters of the place you want to go to.<br>In the 3D interface things/services may be packed more, so you don't have to navigate so much. |

## 11.3. Raw test results

### 11.3.1. Observations

The most interesting observations are written down in the table below together with the subjects that showed the behaviour. Behaviour that occurred with the different interfaces or that independent of the interface type is written down under 'General observations'.

| **General observations** | |
|---|---|
| **Observation** | **Observed** |
| The name "LightHouse" for a service is confusing because it is not a place. | 16 |
| Consciously reasons where the Text Editor Service (a 'non-physical' service) could reside, and comes up with the right answer (Office). | 123456 |
| Correctly searches for a clock when asked to check the time, and goes to the places where he/she would normally expect a clock | 123456 |
| Troubles understanding the concept of a Service's User Interface searching for other services to cooperate with (e.g. the interface for the digital camera that can scan the local network for displaying services). | 13 |
| Expects a 'hall place' from where to control the light in the hall | 156 |
| Got to get used to the idea that you can go both ways in when using one service from within another (e.g. when displaying pictures from the digital camera on the TV, the subject could start with the camera and find the TV, or the other way around). | 36 |

| | |
|---|---|
| Confuses reality with virtuality (tries to find a computer in the Services Sphere and then realizes he/she sits behind one and tries to use that one) | 26 |
| Confused by the concept of a HotSpot place | 1 |

| Text-based interface | |
|---|---|
| **Observation** | **Observed** |
| Uses help commands a lot in the beginning | 126 |
| Tries to use commands from existing command line tools (DOS/UNIX) | 25 |
| Quickly learns to navigate around after a slow start | 123456 |
| The use of help commands diminishes quickly | 123456 |
| Thinks the commands are awkward | 6 |
| Case-sensitivity is confusing. | 123456 |
| Spelling errors are not easily realized | 16 |
| Can't find local services | 3 |
| Is initially confused by the separation between services and places | 4 |
| Finds it confusing that places are listed under services as well | 6 |
| Would like to automatically see all available places and services | 2 |
| Expects a text-based UI instead of a graphical UI for the services | 5 |
| Wishes for more metadata about the services to find out what they are before launching them | 5 |
| Thinks every service is a place: 'go service' instead of 'go place' | 1246 |
| Tries to launch a place | 3 |
| Is confused by the long retrieval time, thinks the browser crashed | 6 |
| Tries to go directly to a place while there is no link between the two places (e.g. from HotSpot to Kitchen) | 2456 |

| 2D interface | |
|---|---|
| **Observation** | **Observed** |
| No problems getting started | 156 |
| Has learned from experience with text-based interface | 156 |
| Becomes proficient very fast | 156 |
| No more errors of trying to go from one place to another while there is no link between them | 156 |
| Finds 'history list' confusing | 5 |

| | |
|---|---|
| Uses 'HotSpot' and 'Local' button without wondering what they are for | 1 |

| **3D interface** | |
|---|---|
| **Observation** | **Observed** |
| Has to get used to using a keyboard and a mouse for navigating 3D environments | 6 |
| Uses the keyboard and mouse for navigating without any problems | 1 |
| Finds the interface more intuitive but too slow. | 16 |
| Gets annoyed by how long it takes to navigate between services | 156 |
| Gets confused by the abstract space that contains all services that have no 'physical' counterpart (MetaVerse) | 156 |
| Finds it irritating that he has to click an object to see if, and if so, what services it provides. A computer can for instance provide many services (sound, display, calculation etc.), while a bookshelf might provide no service at all. | 15 |
| Misses a listing of all services in a place | 5 |
| Wishes for a map of the house to jump directly to a place | 5 |

### 11.3.2. Questionnaires

| **Tasks given to the users** |
|---|
| Change the temperature of the refrigerator in your kitchen.<br><br>Check the web camera in your living room<br><br>See what services are available on the local network.<br><br>Launch a text editor.<br><br>Check the time.<br><br>Turn on the light in the hall of your house.<br><br>Launch the digital camera and display the images on the TV.<br><br>Print an image. |

| **Quality of use – Text Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| This software responds too slowly to inputs. | | | 123456 |
| I would recommend this software to my colleagues. | 25 | 1346 | |
| The instructions are helpful. | 2346 | 5 | 1 |
| The software has at some time stopped unexpectedly. | 346 | | 215 |
| Learning to operate this software initially is full of problems. | | 13 | 2456 |
| I sometimes don't know what to do next with this software. | 123 | 45 | 6 |
| I enjoy my sessions with this software. | 345 | 16 | 2 |
| I find that the help given by this software is not very useful. | 2 | 1 | 3456 |
| If this software stops, it is not easy to restart it. | | 1345 | 26 |
| It takes too long to learn the software commands. | 1 | | 23456 |
| I sometimes wonder if I'm using the right command. | 1246 | | 35 |
| Working with this software is satisfying. | 23 | 145 | 6 |
| I feel safer if I use only a few familiar commands or operations. | 3 | 145 | 26 |
| This software seems to disrupt the way I normally arrange my work. | 3 | 12 | 456 |
| Working with this software is mentally stimulating. | 1 | 2345 | 6 |
| There is never enough information on the screen when it's needed. | 1 | 35 | 246 |
| I feel in command of this software when I am using it. | 2356 | 4 | 1 |

| | | | |
|---|---|---|---|
| I think this software is inconsistent. | | 16 | 2345 |
| I would not like to use this software every day. | 36 | 4 | 125 |
| I can understand and act on the information provided by this software. | 123456 | | |
| This software is awkward when I want to do something not standard. | 2 | 14 | 356 |
| Tasks can be performed in a straightforward manner with the software. | 2345 | 1 | 6 |
| Using this software is frustrating. | | 1 | 23456 |
| The software has helped me overcome any problems I had using it. | | 123456 | |
| The speed of this software is fast enough. | 23456 | 1 | |
| I keep having to go back to look at the guides. | 6 | 4 | 1235 |
| It is obvious that user needs have been fully taken into consideration. | 15 | 34 | 26 |
| Sometimes when using this software I have felt quite tense. | | 1345 | 26 |
| The organisation of the menus or information lists seems quite logical. | 123456 | | |
| Learning how to use new functions is difficult. | 1 | 4 | 2356 |
| There are too many steps required to get something to work. | 16 | 4 | 235 |
| It is easy to make the software do exactly what you want. | 2356 | 14 | |
| I will never learn to use all that is offered in this software. | 4 | 13 | 256 |
| The software hasn't always done what I was expecting | 36 | 4 | 125 |
| The software has a very attractive presentation. | | 4 | 12356 |
| It is easy to forget how to do things with this software. | | 4 | 12356 |
| This software occasionally behaves in a way that can't be understood. | | 1 | 23456 |

| | | | |
|---|---|---|---|
| This software is really very awkward. | | 14 | 2356 |
| It is easy to see at a glance what the options are at each stage. | 4 | 1356 | 2 |
| I have to look for assistance most times when I use this software. | 23 | 4 | 156 |

| **Navigation – Text Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| Navigating takes a lot of effort. | 1 | 4 | 2356 |
| I sometimes get lost. | 13 | 45 | 26 |
| It is always clear where in the system you are. | 345 | 16 | 2 |
| It is easy to go from one place to another. | 346 | 2 | 15 |
| I sometimes forget where I am in the system. | 13 | 4 | 25 |
| Navigating is inconsistent. | | 1 | 23456 |
| It’s simple to go forward and backward. | 3456 | 2 | 1 |
| The history list is helpful. | 6 | 12345 | |
| There is not enough information what you can do in a certain place. | 125 | 4 | 36 |
| I found the geographical metaphor confusing. | 1 | 345 | 26 |
| The way services are organized is intuitive. | 12356 | 4 | |
| The way services are organized is useful. | 2356 | 14 | |
| Finding and using local services was hard. | | 14 | 2356 |
| Making 2 services work together is difficult. | 12 | 346 | 5 |
| The information about a service sufficient to judge whether it was what I needed? | 56 | 3 | 124 |

| | | | |
|---|---|---|---|
| It was easy to find out what a service could do | 56 | 3 | 124 |
| The name and location is sufficient for finding out what a service does | | 36 | 1245 |

| **Quality of use – 2D Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| This software responds too slowly to inputs. | 1 | | 2345 |
| I would recommend this software to my colleagues. | 2345 | 1 | |
| The instructions are helpful. | 134 | 5 | 2 |
| Learning to operate this software initially is full of problems. | | | 12345 |
| I sometimes don't know what to do next with this software. | 2 | | 1345 |
| I enjoy my sessions with this software. | 12354 5 | | |
| I find that the help given by this software is not very useful. | 2 | | 1345 |
| It takes too long to learn the software commands. | 3 | | 1245 |
| I sometimes wonder if I'm using the right command. | 123 | | 45 |
| Working with this software is satisfying. | 5 | 2 | 134 |
| This software seems to disrupt the way I normally arrange my work. | | 235 | 14 |
| Working with this software is mentally stimulating. | 45 | | 123 |
| There is never enough information on the screen when it's needed. | | 135 | 24 |
| I feel in command of this software when I am using it. | 245 | | 13 |

| | | | |
|---|---|---|---|
| I think this software is inconsistent. | | | 12345 |
| I would not like to use this software every day. | 3 | 1 | 245 |
| I can understand and act on the information provided by this software. | 145 | | 23 |
| Tasks can be performed in a straightforward manner with the software. | 1345 | | 2 |
| Using this software is frustrating. | | 1 | 2345 |
| The software has helped me overcome any problems I had using it. | 1 | 345 | 2 |
| The speed of this software is fast enough. | 34 | 5 | 12 |
| I keep having to go back to look at the guides. | 12 | | 345 |
| It is obvious that user needs have been fully taken into consideration. | 5 | 4 | 123 |
| Sometimes when using this software I have felt quite tense. | | 3 | 1245 |
| The organisation of the menus or information lists seems quite logical. | 1345 | | 2 |
| Learning how to use new functions is difficult. | | 3 | 1245 |
| There are too many steps required to get something to work. | | | 12345 |
| It is easy to make the software do exactly what you want. | 345 | 1 | 2 |
| I will never learn to use all that is offered in this software. | | 3 | 1245 |
| The software hasn't always done what I was expecting. | | | 12345 |
| The software has a very attractive presentation. | 234 | 5 | 1 |
| It is easy to forget how to do things with this software. | 1 | | 2345 |
| This software occasionally behaves in a way that can't be | | | 12345 |

| | | | |
|---|---|---|---|
| understood. | | | |
| This software is really very awkward. | | | 12345 |
| It is easy to see at a glance what the options are at each stage. | 12345 | | |
| I have to look for assistance most times when I use this software. | | | 12345 |

| **Navigation – 2D Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| Navigating takes a lot of effort. | | | 12345 |
| I sometimes get lost. | 1 | | 2345 |
| It is always clear where in the system I am. | 45 | 3 | 12 |
| It is easy to go from one place to another. | 2345 | | 1 |
| I sometimes forget where I am in the system. | 1 | | 2345 |
| Navigating is inconsistent. | | | 12345 |
| It's simple to go forward and backward. | 1234 | 5 | |
| The history list is helpful. | 243 | 13 | 5 |
| There is not enough information what you can do in a certain place. | 23 | 5 | 14 |
| I found the geographical metaphor confusing. | 23 | 1 | 45 |
| The way services are organized is intuitive. | 145 | | 23 |
| The way services are organized is useful. | 45 | 1 | 23 |
| Finding and using local services was hard. | | | 12345 |
| Making 2 services work together is difficult. | 23 | 4 | 15 |
| The information about a service is sufficient to judge whether it is what I need. | 14 | 5 | 23 |
| It was easy to find out what a service can do. | 1245 | | 3 |

| | | | |
|---|---|---|---|
| The name and location is sufficient for finding out what a service does. | 15 | 4 | 2 |

| **Quality of use – 3D Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| This software responds too slowly to inputs. | 2345 | | 1 |
| I would recommend this software to my colleagues. | 2 | | 1345 |
| The instructions are helpful. | | | 12345 |
| Learning to operate this software initially is full of problems. | 25 | 3 | 14 |
| I sometimes don't know what to do next with this software. | 1235 | | 4 |
| I enjoy my sessions with this software. | 12 | 3 | 45 |
| I find that the help given by this software is not very useful. | 125 | | 34 |
| It takes too long to learn the software commands. | | 5 | 1234 |
| I sometimes wonder if I'm using the right command. | 1 | 5 | 234 |
| Working with this software is satisfying. | 12 | 3 | 45 |
| This software seems to disrupt the way I normally arrange my work. | 45 | 3 | 12 |
| Working with this software is mentally stimulating. | 12 | 3 | 45 |
| There is never enough information on the screen when it's needed. | 135 | 4 | 2 |
| I feel in command of this software when I am using it. | 2 | 1 | 345 |
| I think this software is inconsistent. | | 45 | 123 |
| I would not like to use this software every day. | 1345 | | 2 |
| I can understand and act on the information provided by this software. | 12 | 3 | 45 |
| This software is awkward when I want to do something not standard. | 13 | 5 | 24 |
| Tasks can be performed in a straightforward manner with | 2 | | 1345 |

| | | | |
|---|---|---|---|
| the software. | | | |
| Using this software is frustrating. | 345 | | 12 |
| The software has helped me overcome any problems I had using it. | 2 | 1 | 345 |
| The speed of this software is fast enough. | | 2 | 1345 |
| I keep having to go back to look at the guides. | | 1235 | 4 |
| It is obvious that user needs have been fully taken into consideration. | | 24 | 135 |
| Sometimes when using this software I have felt quite tense. | 45 | 2 | 13 |
| The organisation of the menus or information lists seems quite logical. | 24 | 135 | |
| Learning how to use new functions is difficult. | 5 | 1 | 234 |
| There are too many steps required to get something to work. | 1235 | | 4 |
| It is easy to make the software do exactly what you want. | 2 | | 1345 |
| I will never learn to use all that is offered in this software. | | | 12345 |
| The software hasn't always done what I was expecting. | 123 | 5 | 4 |
| The software has a very attractive presentation. | 1234 | 5 | |
| It is easy to forget how to do things with this software. | | | 12345 |
| This software occasionally behaves in a way that can't be understood. | 2 | 5 | 134 |
| This software is really very awkward. | 5 | | 1234 |
| It is easy to see at a glance what the options are at each stage. | | | 12345 |
| I have to look for assistance most times when I use this software. | | 5 | 1234 |

| **Navigation – 3D Interface** | **Agree** | **Undecided** | **Disagree** |
|---|---|---|---|
| Navigating takes a lot of effort. | 12345 | | |
| I sometimes get lost. | 15 | | 234 |
| It is always clear where in the system you are. | 1234 | | 5 |
| It is easy to go from one place to another. | 2 | | 1345 |
| I sometimes forget where I am in the system. | 5 | | 1234 |
| Navigating is inconsistent. | | | 12345 |
| It's simple to go forward and backward. | 134 | | 25 |
| There is not enough information what you can do in a certain place. | 1235 | | 4 |
| I found the geographical metaphor confusing. | | 15 | 234 |
| The way services are organized is intuitive. | 124 | | 35 |
| The way services are organized is useful. | 24 | 1 | 35 |
| Finding and using local services was hard. | 2345 | | 1 |
| The information about a service sufficient to judge whether it is what I need. | | | 12345 |
| It was easy to find out what a service can do. | 14 | | 235 |
| The location is sufficient for finding out what a service does. | 4 | | 1235 |

### 11.3.3. Open questions

What follows are the unedited answers to the open questions.

| **Respondent 1** |
| --- |
| Do you feel that navigating services is very different from navigating documents (e.g. Web browsing), why (not)?<br><br>No, strong similarities whith the way documents are ordered in an "old fashion" tree view.<br><br>If so, what are the most important differences for you? |
| Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?<br><br>text: When knowing the commands, the navigation goes more rapidly. |
| Name the advantage of each types of interface *for navigating services*:<br><br>Text-based:<br>Quick jumps from one place to another (eg. Room to Room)<br>What u see is what you can use (eg. No 'useless' objects like walls, tables etc. )<br><br>2D:<br>Very clear to see where you can go<br>More intuitive interface<br><br>3D:<br>Very intuitive for locating services or places<br>Fun to use |

| |
|---|
| Name the disadvantage of each types of interface *for navigating services*:<br><br>Text-based:<br>Harder to learn (have to learn commands)<br>less intuitive<br><br>2D:<br>A lot of mouseclicks are nessecery to get where you want to go.<br><br>3D:<br>Slow navigation<br>to much ubiquitus objects (had to mouseclick the table just to be sure there was no service connected to it) |
| Do you have suggestions for other types of interfaces that you would like?<br><br>"first person shooter" like (eg. Quake) whith quick jumps to other places.<br>Some sort of augmented **virtual** reality. |
| Describe the mental picture of where you were while navigating? |
| Do you think there are better ways to organize the services? E.g. by time or activity.<br><br>Depends completely on the service itself. |
| Can you think of another metaphor? Would that be a better metaphor?<br><br>NO |
| Is the geographical metaphor more useful with one type of interface compared to the others?<br><br>Yes |
| Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?<br><br>Yes, the 3D gives more visual information location, service(device) etc. |
| Can you think of additional ways for services to convey information about what they offer?<br><br>augmented **virtual** reality |

**Respondent 2**

Do you feel that navigating services is very different from navigating documents (e.g. Web browsing), why (not)?

Yes: the way services are organized and so the way to find them are different from documents. It uses a different metaphor. The same naming of places and services can be confusing. It isn't clear what the difference is, if u just look at it. Documents are mostly organized on their relation in meaning, on the information they contain, while the services (which don't have to contain information themselves, but they have a certain function) are oriented more towards their function and the location and occasion you would use them. This differs per person, so it can be that the organisation for one person is totally different from the other. I think it's therefore more difficult to organize services on a universally understandable way to users then documents. The other thing is that they are more abstract. People unfamiliar with a service or concept probably won't even notice it, or what its for.

For the 3d version it ways indeed a totally different way of navigating, more like finding your way in a building you enter for the first time. This exploring way of finding is fun , especially if u are unaware what is there to encounter. This makes it having a high 'fun to do' level. TO get things done u always do, and you wnat to get done quickly because u have to (like cleaning the house :-) this interface is less suitable. A combination of all the interface would be preferable. This 3d interface is in my eyes the ideal tool to learn complex task, or a good way to teach children.

If so, what are the most important differences for you?

The organisation of documents depends on the context information they provide and give me clues of what I'll find on the next click. Services are more clueless and depend more on my general knowledge of similar services and concepts of the real world.

Totally grafically, instead of text based. So it would be multilingual and also workable for illiterate. Because of it thurst of computer resources it isn't as 'snappy' in use as regular browserS, navigational devices. This can be a pluspoint as people tend to remember locations better as they see how they got there, so to say the context info. (just like u are walking in a large supermarket. U might not remember the exact location, but recognize the route towards the goal). Part from that there can be displayed a lot of information graphically, in comparison to text / button based screens.

| |
|---|
| Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?<br><br>To use dayly, i like the 2d interface the most, because it lets u quickly, but clearly, do your job. For more complex situations, and to make a deeper impression (for example to memorize the route through a building or a city) i would use the 3d interface. It's the easiest way to remember much information as it works on recognizing instead of remembering. Only on a real bare and stripped device, or with a lack of resources i'd use the commandline tool. It can do the job, but there's no fun in doing it. |
| Name the advantage of each types of interface *for navigating services*:<br><br>Text-based:<br>It's a quick way of using services. Problem can be that it doesn't offer the ability to explore easily (if I saw a list of services and I wouldn't be able to recognize what something was, I would be curious and go there to explore).<br>The things that where confusing to me (being a visually oriented Windows user) were that the commands the few basic UNIX or DOS commands I knew (such as 'ls' or 'dir' to see a listing etc.).<br>Also I'd searched for some metadata on services, a short functional description of the service and where you can find it.<br>Requires low resources, usable on very old devices/ small devices. (Requires little window space, mouse)<br><br>2D:<br>The combination of speed and pleasant, clear presentation. More options displayed once, so better to judge what choice to make for a task.<br>The 2d browser would be even clearer when every place could be displayed in a tree. Showing the hierarchy and relation between places at once. Within this tree, at the branches, showed by a explicit different symbol, the services could be shown (so the relation between places and services would be visible, on multiple levels at once, and the relations between places themselves and services themselves would be visible). It could be seen as a small, slim, service browser, the control centre from which to coordinate your tasks.<br><br>3D:<br>creates a good impression of relations between services, locations.it is the ideal interface to show very much information (if it is visualizable). Works good if you are curious, like to explore. Very low learning curve, only the navigation itself. But using the services is like doing things in everyday live. |

| |
|---|
| Name the disadvantage of each types of interface *for navigating services*:<br><br>Text-based:<br>It doesn't stimulate to use it for fun. It's sober and I would only use it for strict necessary things, not for the amusement or joy.<br>Lack of orientation points. Where am i, how did i cam here, what can i do here. It requires a proactive approach. U have to actively wanderer around to see what's there<br>It think it will be a powerful interface on the more restricted devices with less resources (small screen, low bandwidth), but these devices would definitely need a keyboard, because point 'n click isn't possible.<br><br>2D:<br>It uses more space then necessary. Because it looks different from familiar applications, its hard to rely on previous software experience. U still have to figure out what everything does, and why its displayed as is. Its still rather basic, and at a first glance seems lacking of functionality.<br><br>3D:<br>drain on resources. You don't have a total view of all the available services. Usable only by People who are experienced mouse users, and used to this 'real world navigation'. |
| Do you have suggestions for other types of interfaces that you would like?<br><br>A combination of the three interfaces would also be very strong combination. U could wander around in a place and click on a spot to go directly to a 2d interface to get a overview, or get things done. In a 2 d interface, a switch to a 3d environment would help in the understanding of the offered services, like a sort of guide. |
| Describe the mental picture of where you were while navigating?<br><br>With every service i'd use i would think if i already knew a similair service or device and in which context (software and physical environment) i used it. Thsi would give me clues where to expect something.<br>With the 3d interface the mental modal i had was constantly corrected by the feedback i got from the virtual world i was in. So it was the closest match to my mental model. |
| Do you think there are better ways to organize the services? E.g. by time or activity.<br><br>It depends on the service, if it has a direct 'pointable' physical equivalent it is the best way I guess, it becomes harder for abstract ones. Maybe abstract categories as |

| |
|---|
| ‘finance’ or something for all the banking services would be an approach.<br>Being able to appoint and reorganize the services, and a total inventory/list of all services at once would be helpful.<br><br>It think the best way to organize it is to ask the user to do so, or to adopt to the users needs (if it uses certain services often together, after each other there should be some connection between them for the user). I’d love to get offered a default global organisation, which i could customize towards my personal needs. This can differ on occasions. When on holiday i would use different services then at the office.<br>In the 3d interface i think this is the best way to organize things. It closely matches the knowledge people already have of the world, (at least people familiar with the presented, probably western oriented world) |
| Can you think of another metaphor? Would that be a better metaphor?<br><br>The more function oriented metaphor (can’t figure out a name yet), in which device settings were grouped (devices for fun, such as tv, photo,vcr, radio, games, chatting, devices to get things done; fridge, microwave, cv and devices to communicate : mail, phone, cam, devices to support me in things: calculator, textpad, spreadsheet, search tool) there is probably lot of overlap but it depends on the situation in which services would be grouped and in which not. I would cal this situational offering. |
| Is the geographical metaphor more useful with one type of interface compared to the others?<br><br>Yes, more physical services are more familiar in general and the same for everybody whereas the more abstract services can differ a lot per person, his background, location, culture, experience etc.<br><br>It totally suites the 3d interface, while less the 2d interface and not the commandline interface. But people familiar with all interfaces have the advantage of seeing relations between them. (places seen in 3d can be recognized in the commandline) |
| Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?<br><br>Yes, 3d is the most intuitive, as u immediatly see what things are meant to do or to be. It can be tricky however woth more abstract services, but visual clues like then text on an object can help  alot, just like in the ‘real world’ |
| Can you think of additional ways for services to convey information about what they offer? |

| |
|---|
| Comments<br><br>Something that can be confusing for users is the lack of physical /local storage of things. People should be made aware that documents are stored for them somewhere, but they also should have the option to explicitly store things on their personal PC, device (personal data on a CD), so they had the feeling of being in control and people would be able to adopt at their own rate to this new way of thinking of using services and information.<br><br>In the 3d interface things/services may be packed more, so u don't have to navigate so much. |

| **Respondent 3** |
|---|
| Do you feel that navigating services is very different from navigating documents (e.g. Web browsing), why (not)?<br><br>No, I think that despite the different terminology there will always be a similarity with Web browsing. This is probably due to the fact that both have to do with using the Internet.<br><br>If so, what are the most important differences for you? |
| Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?<br><br>2D: it's fast, has a good overview of the services (if well designed!), recognizable and easy to learn. It's also easy to accommodate it, for example for different languages. |

| |
|---|
| Name the advantage of each types of interface *for navigating services*:<br><br>Text-based:<br>Only when you are a fast learner of commands. It is not good for inexperienced computer users.<br><br>2D:<br>Easy to use by all types of users.<br><br>3D:<br>Good for demo's and children.<br>It will increase the learning time, which will cost you users in the end.<br>It is not good if you want to quickly do something.<br>Good for first-time users. |
| Name the disadvantage of each types of interface *for navigating services*:<br><br>If the subjects stays vague (digital services on the internet), no type of interface will help the user in getting a clear picture of what the system offers.<br><br>For computerfreaks text-based interfaces are always faster than 2D interfaces.<br><br>3D:<br>Slow<br>Difficult to design<br>Expensive |
| Do you have suggestions for other types of interfaces that you would like?<br><br>No idea, maybe that iMode or PDA's offer more opportunities. |
| Describe the mental picture of where you were while navigating?<br><br>Mentally I was where I was physically, right in front of a PC. |
| Do you think there are better ways to organize the services? E.g. by time or activity.<br><br>A geographical metaphor is not fitting, maybe a personal metaphor is more useful. |
| Can you think of another metaphor? Would that be a better metaphor? |

| |
|---|
| Is the geographical metaphor more useful with one type of interface compared to the others?<br><br>It might be useful, but only up 'till a certain point. At one point the variables that make up the geographical metaphor vary too much to be of any use anymore. |
| Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?<br><br>Depends on the user. Experienced computer users will be know how to use all 3 types of interface. A beginner might prefer the 3D interface.<br><br>2D offers: recognition/reliance/consistency/control/effectiveness/flexibility |
| Can you think of additional ways for services to convey information about what they offer?<br><br>Arrange *icons*, that are unique for each service, by hand. |
| Comments<br><br>The technique might be different, but don't try to reinvent the wheel as far as the interface goes. It is still navigation and the basic laws for that haven't changed. |

| **Respondent 4** |
|---|
| Do you feel that navigating services is very different from navigating documents (e.g. Web browsing), why (not)?<br><br>No, navigating  services is with a view basic rules intuitive, just like web browsing.<br>If so, what are the most important differences for you? |
| Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?<br><br>2d; most easy (quick) to handle. |

| |
|---|
| Name the advantage of each types of interface *for navigating services*:<br><br>Text-based:<br><br>2D:<br>navigating is very easy (especially compared to 3d, but I think playstation-kids won't agree with me)<br><br>3D:<br>you can easily see were you are in the system |
| Name the disadvantage of each types of interface *for navigating services*:<br><br>Text-based:<br>Slow navigating because of the amount of typing that it requires.<br><br>2D:<br>none<br><br>3D:<br>slow navigating |
| Do you have suggestions for other types of interfaces that you would like?<br><br>no |
| Describe the mental picture of where you were while navigating?<br><br>I was thinking of a tree-structure with the text and 2 D interface, with 3D I was thinking of moving in a picture. |
| Do you think there are better ways to organize the services? E.g. by time or activity.<br><br>3d joystick or something like it to make navigating easy, or a combined txt/3d, for example you are in the living room and want to navigate to the bathroom, while using the right mouse button, there comes in a text field in which you can enter the first tree characters of the place you want to go and you go there immediately |
| Can you think of another metaphor? Would that be a better metaphor?<br><br>no |

| |
|---|
| Is the geographical metaphor more useful with one type of interface compared to the others?<br><br>No to me that the difference between them and the difference in functionality is not related to the metaphor. |
| Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?<br><br>No |
| Can you think of additional ways for services to convey information about what they offer?<br><br>Location sets the expectations to go and look for a service and what you think it could do in advance. |

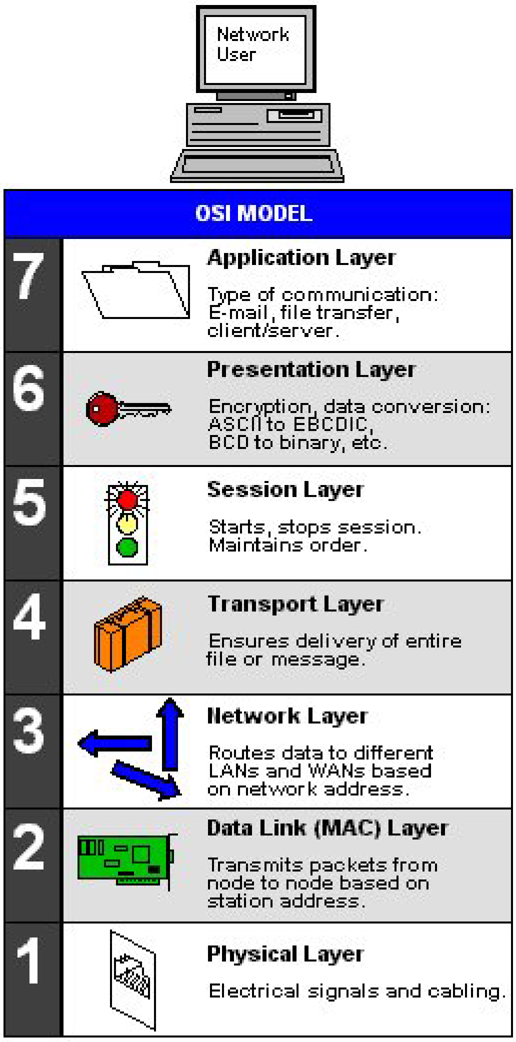
| **Respondent 5** |
|---|
| Do you feel that navigating services is very different from navigating documents (e.g. Web browsing), why (not)?<br><br>In the 2D and text based interface it is very similar. But the 3D interface was much more difficult to navigate.<br><br>If so, what are the most important differences for you?<br><br>The most important difference is the amount of time it takes to get from one service to another in the 3D interface. Furthermore the available services at a place was not 'listed' in the 3D interface as is a list of documents in a webpage. Antoher important difference is not being able to search for a service in the interface, in a way that you can when browsing documents. |
| Which of the three types of interfaces (text, 2D or 3D) do you like best for navigating services, and why?<br><br>By far the 2D interface. It was very fast to navigate, had a much better overview than both the text based and 3D interface. When using the text based and the 2D interface, one creates a 'mental picture' of the place that is navigated through. In the 3D interface this picture is already made for the user, and that 'preconstructed picture' might differ very much from the users own 'mental picture', which only |

| |
|---|
| confuses the user. |
| Name the advantage of each types of interface *for navigating services*:<br><br>Text-based:<br>One have a feeling of complete control when using the text based interface<br><br>2D:<br>Very good overview of the available services, and at the sametime very fast to navigate.<br><br>3D:<br>Nice graphics and that is about it! |
| Name the disadvantage of each types of interface *for navigating services*:<br><br>Text-based:<br>Have to do a lot of typing which decreases the speed of navigating<br><br>2D:<br>No real disadvantage<br><br>3D:<br>Already stated above |
| Do you have suggestions for other types of interfaces that you would like?<br><br>No. |
| Describe the mental picture of where you were while navigating?<br><br>I had a picture of the physical surroundings when navigating the services using the text and 2D interface. With the 3D interface I had no mental picture! It was already created for me.<br><br>Actually had a feeling of physically walking around in the places. Associates placement of services to physical places. |
| Do you think there are better ways to organize the services? E.g. by time or activity.<br><br>Do not think so. But it is hard to tell if this kind of 'physical mapping' would apply well to all kinds of services. E.g. where would 'joke of the day service' be located? Maybe one should be able to switch between types of ways to organize services.<br><br>Maybe better organized by use. I might want to use my text editor service in my |

| |
|---|
| kitchen, toilet etc. Therefore the goegraphical mapping is not always logical. But if having to choose one, I would regard this as being the best. |
| Can you think of another metaphor? Would that be a better metaphor?<br><br>No. |
| Is the geographical metaphor more useful with one type of interface compared to the others?<br><br>I find it more useful in the text-based and 2D version, since in the 3D browser I do not really find it a metaphor. |
| Is there a difference between the types of interfaces in how easy it is to understand what types of services are being offered?<br><br>Yes. When using the 3D interface one would need to have before hand knowledge of a service. It would have been funny trying to use the 3D interface first! I think that would have resulted in an even harsher judgement! |
| Can you think of additional ways for services to convey information about what they offer?<br><br>No. |

## 11.4. OSI model overview

The following overview is largely based on information found in the Computer Desktop Encyclopedia (Computer Desktop Encyclopedia, 1998).




**Application - Layer 7**

This top layer defines the language and syntax that programs use to communicate with other programs. The application layer represents the purpose of communicating in the first place. For example, a program in a client workstation uses commands to request data from a program in the server. Common functions at this layer are opening, closing, reading and writing files, transferring files and e-mail messages, executing remote jobs and obtaining directory information about network resources.

**Presentation – Layer 6**

When data is transmitted between different types of computer systems, the presentation layer negotiates and manages the way data is represented and encoded. For example, it provides a common denominator between ASCII and EBCDIC machines as well as between different floating point and binary formats. Sun’s XDR and OSI’s ASN.1 are two protocols used for this purpose. This layer is also used for encryption and decryption.

**Session – Layer 5**

Provides coordination of the communications in an orderly manner. It determines one-way or two-way communications and manages the dialogue between both parties; for example, making sure that the previous request has been fulfilled before the next one is sent. It also marks significant parts of the transmitted data with checkpoints to allow for fast recovery in the event of a connection failure. In practice, this layer is often not used or services within this layer are sometimes incorporated into the transport layer.

**Transport – Layer 4**

The transport layer is responsible for overall end-to-end validity and integrity of the transmission. The lower data link layer (layer 2) is only responsible for delivering packets from one node to another. Thus, if a packet gets lost in a router somewhere in the enterprise Internet, the transport layer will detect that. It ensures that if a 12MB file is sent, the full 12MB is received. "OSI transport services" include layers 1 through 4, collectively responsible for delivering a complete message or file from sending to receiving station without error.

**Network – Layer 3**

The network layer establishes the route between the sending and receiving stations. The node-to-node function of the data link layer (layer 2) is extended across the entire Internet, because a routable protocol contains a network address in addition to a station addresses.

This layer is the switching function of the dial-up telephone system as well as the functions performed by routable protocols such as IP, IPX, SNA and AppleTalk. If all stations are contained within a single network segment, then the routing capability in this layer is not required.

**Data link – Layer 2**

The data link is responsible for node-to-node validity and integrity of the transmission. The transmitted bits are divided into frames; for example, an Ethernet, Token Ring or FDDI frame in local area networks (LANs). Layers 1 and 2 are required for every type of communications.

**Physical layer – Layer 1**

The physical layer is responsible for passing bits onto and receiving them from the connecting medium. This layer has no understanding of the meaning of the bits, but deals with the electrical and mechanical characteristics of the signals and signalling methods. For example, it comprises the RTS and CTS signals in an RS-232 environment, as well as TDM and FDM techniques for multiplexing data on a line. SONET also provides layer 1 capability.